Army Materiel Systems Analysis Activity

AMSAA SPECIAL REPORT NO. SR-2017-08

Force -on -Force Modeling with Formal Task Structures and Dynamic Geometry

MARCH 24, 2017

DISTRIBUTION STATEMENT A: Approved for public released: Distribution unlimited.

U.S. ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY

ABERDEEN PROVING GROUND, MARYLAND 21005-5071

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
24/03/2017

2. REPORT TYPE
Final Report

3. DATES COVERED *(From - To)*
08-03-2016 – 01-03-2017

4. TITLE AND SUBTITLE
Force-on-Force Modeling with Formal Task Structures and Dynamic Geometry

5a. CONTRACT NUMBER
W911NF-11-D-0001 DO# 0362

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)
Britt E. Bray, John J. Nelson, Beth S. Ward, Paul H. Deitz, Russell Dibelka, James Hunt, Martin Wayne

5d. PROJECT NUMBER
N/A

5e. TASK NUMBER
TCN 16-004

5f. WORK UNIT NUMBER
N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Materiel Systems Analysis Activity
ATTN: AMXAA-D
Aberdeen Proving Ground, MD 21005-5068

8. PERFORMING ORGANIZATION REPORT NUMBER
SR -2017 -08

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Deputy Under Secretary of the Army, Test & Evaluation
2350 Crystal Drive, Suite 8000
Arlington, VA

10. SPONSOR/MONITOR'S ACRONYM(S)
DUSA-TE

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A: Approved for public released. Distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This study demonstrated the application of a formal task framework, derived using the MMF methodology to structure a complex mission. It further demonstrated the integration of effects from a range of kinetic interactions, reliability failures and repair events as produced by a systems vulnerability model (MUVES) using an Application Program Interface (API) to pass data between the contractor developed FoF model and MUVES.

15. SUBJECT TERMS
Mission and Means Framework, Dynamic Geometry, Capability Analysis, Simulation, MUVES, ExtendSim

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPONE NUMBER (*Include area code)*
U	U	U	SAR	62	

Scientific Services Program

The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The purpose of this pilot project was to demonstrate a form of Force-on-Force (FoF) modeling using both formal tasks and dynamic geometry at the unclassified level through application of the Missions and Means Framework (MMF) levels and operators. The specific application and the application methodology was intended to support a combined developmental testing (DT) and operational testing (OT) strategy for selected systems under test per the mission of ATEC Army Evaluation Center (AEC). And beyond testing, this singular integrating formalism has significant ramifications across a broad group of requirements, research, test, training, and analytic activities, all of which are identically mirrored in this conceptual model

This work promises to support Army Materiel Systems Analysis Activity (AMSAA), Army Research Laboratory Survivability/Lethality Analysis Directorate (ARL SLAD), and the ATEC missions to assess system effectiveness by providing the logical and executable architecture to integrate system and operational analyses via FoF simulations (e.g., OneSAF).

Specific objectives for the demonstration were to:

1. Prepare for the Test & Evaluation (T&E) analysis process by:

a. Identifying mission essential task (METs) and supporting tasks correlated to operational requirements for the Armored Multi-Purpose Vehicle (AMPV).

b. Develop the method for linking appropriate METs and supporting tasks to capability definitions of AMPV mission variants.

2. Develop a MMF Dynamic Assessment Suite model that instantiates elements of the Missions and Means Framework (MMF) to:

a. Simulate ballistic interaction and reliability effects on a tactical unit equipped with AMPV mission variants for a select vignette in the context of an unclassified operational scenario.

b. Simulate information interactions and resulting situational understanding and decision-making effects on a tactical unit equipped with AMPV mission variants for a select vignette in the context of an unclassified operational scenario.

3. Execute the T&E assessment:

a. Augment current vulnerability/lethality (V/L) analyses with “capability to task”, and “task to mission” assessment based on evaluation of performance at the platform/system level and resulting performance and mission effectiveness at the collective (unit) level over time. Evaluation will explicitly account for platform degradation caused by reliability failures and/or ballistic interactions by incorporating dynamic system geometry and task cycles during test.

b. Augment current information collection and sensor effectiveness analyses with “capability to task” and “task to mission” assessment based on evaluation of performance at the sensor/information source level and resulting performance and mission effectiveness at the collective (unit) level over time. Evaluation will explicitly account for sensor and information source capabilities given varying sets of conditions based on operational context.

c. Identify, for subsequent “full-view” analysis, which components/subsystem cause risk for specific tasks in selected vignettes.

4. Demonstrate the capability to integrate results of separate and distinct developmental test events within the mission thread(s) and incorporate into the overall evaluation of the system and system-of-systems (SoS).

5. Demonstrate derived platform capability metrics vice a standard damage assessment list (SDAL) approach with lumped-parameter "utility" metrics.

Our general approach to this effort consisted of the following major activities:

1. **Research.** We focused our research efforts on the following areas:

a. Obtain and digest as much information as possible on operational requirements for the AMPV as the designated system of interest for the demonstration.

b. Review previously published reports, briefings and products related to MMF application in general and on application to platform-level systems analysis.

c. Select and then thoroughly read a TRAC published standard scenario for use as the source of operational context for the study.

d. Locate and review the most up to date doctrinal literature sources for the Army Operations Process, (including the Military Decision Making Process (MDMP)), as well as doctrine and Tactics, Techniques and Procedures (TTP) for the Armored Brigade Combat Team (ABCT) and its subordinate units down to Platoon level. See References section for a complete listing of doctrinal literature sources.

2. **Mission Specification.** Our contractor team from Morris, Nelson & Associates, LLC applied the following process to complete the mission specification portion.

a. **Scenario specification and refinement.** We selected the most relevant portions of the Multi-Level Scenario (MLS) module 2.0 Scenario and captured the associated text and graphics for later reference and to serve as a starting point for development of operational vignettes at the Combined Arms Battalion (CAB), Company/Team (CO/TM) and Platoon levels.

b. **MDMP application.** MLS 2.0 included specific task and purpose statements along with operational graphics for the 7th Heavy Brigade Combat Team (HBCT), 7th Division, IXth Corps along with operational graphics depicting the mission tasks assigned to its subordinate CABs. We initiated MDMP analysis on those products to extend the MLS 2.0 to develop operational vignettes in the form of Course of Action (COA) statements and sketches at the CAB, CO/TM, and Platoon levels. These products were generated for the Extended Tactical Movement, Deliberate Attack and Exploitation, and Deliberate Attack in an Urban Environment missions from the AMPV Operational Mode Summary/Mission Profile (OMS/MP). Similar products for the Irregular Warfare (IW) and Peace Operations portion of the mission profile were deferred due to a lack of sufficient time and inability to incorporate them in the model due to the modeling team's focus on resolving the Application Program Interface (API) issues. Completing these two portions will be the priority for the mission specification portion of planned follow on efforts focused on Information requirements and Information sources.

c. **MMF application.** Using the kinetic MMF Formal Process Flow Diagram as a guide, we applied the MMF to analyze relevant portions of MLS 2.0 and the operational vignettes to generate MMF products for incorporation in the model and support assessment of execution via the model runs. Mission task files and mission threads were also developed to support development of information requirements. While detailed operational vignettes have not yet been developed for the IW and Peace Operations portions there was sufficient mission information in the MLS 2.0 documentation to support development of these products.

3. **System Specification.** Our engineering team from ARL SLAD and AMSAA generated unclassified surrogate system/target models for the AMPV, M1 Tank, M2 Bradley Fighting Vehicle (BFV), and M3 Cavalry Fighting Vehicle (CFV) and loaded them into the Modular Unix-based Vulnerability Estimation Suite (MUVES) model during simulation. MUVES is the Vulnerability model used to specify

the engineered designs of the platforms (to the component level at Level-2) and the effects of ballistic interactions and reliability failures.

4. **MMF Dynamic Assessment Suite model design and development.** Figure 1, below, illustrates a conceptual overview of the MMF Dynamic Assessment Suite model. The model design was intended to automate the MMF levels and operators. ExtendSim© is the simulation development software used to develop the FoF mission specification, execution, and assessment model, hereafter referred to as MMF FoF model, at the core of the MMF Dynamic Assessment Suite. MUVES then returns the state of the platform Level-3 capabilities to the MMF FoF model. Making the conceptual model a reality required a joint contractor/government team effort to develop a working API to enable the exchange of data between the mission specification and system specification models. Numerous unexpected programming bugs and the use of different operating system versions (Windows 10 vs. Windows 7) required hours of unplanned work to overcome causing delays in the development timeline. The net effect of these delays was that the integrated MMF Dynamic Assessment Suite model did not begin working as designed until early February 2017 versus the targeted timeframe of November 2016.

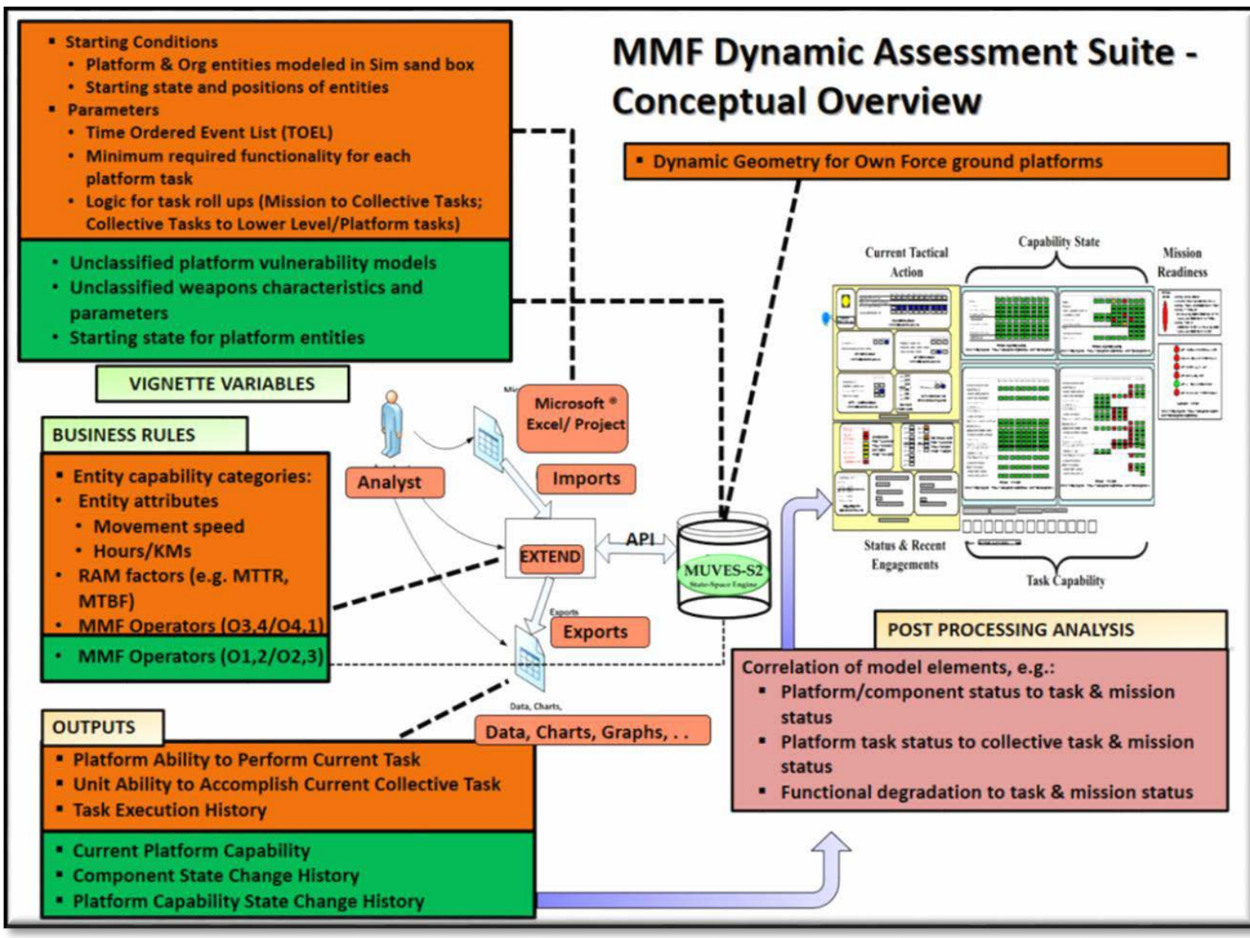


Figure 1 Conceptual View of the MMF Dynamic Assessment Suite model

5. **Model execution and analysis.** Despite delays caused by API development and debugging issues, our modeling team executed 30 complete runs on the Time Ordered Event List (TOEL) for the Deliberate Attack and Exploitation mission. These runs successfully produced results that made sense based on input data and task assessment logic, system data in the MUVES system specifications and

conditions surrounding the vignette interactions. Run results were analyzed and a summary of the analysis results is discussed in Section 5 of this report.

The Force-on-Force Modeling with Formal Task Structures with Dynamic Geometry study successfully achieved the objectives of the study and advanced our ability to model and assess the impact of dynamic changes in platform geometry on mission effectiveness

Results of this study demonstrated the application of the MMF as both a framework and methodology to develop new or modify existing Models and Simulations (M&S) to:

- Apply data from multiple, distributed sources (including test events) to evaluate effectiveness, suitability, and survivability of ground combat systems.
- Account for component and platform level degradation during operational simulation run time by linking a system level survivability and reliability model (MUVES) to a purpose built operational model using shared metrics and an executable integration architecture.
- Generate a formal top-down mission specification for sample operating force organizations equipped with a candidate Army platform (the AMPV)
- Exercise a model environment for scripted and randomized mission threads and interactions.
- Generate and analyze data from 30 separate simulation runs of the mission thread with randomized ballistic, reliability failure and repair interactions.

INTRODUCTION

The stage for this project was set in the 1980s when the Army was engaged in full-up live-fire activities in response to newly established legislation (1). The law required that for each live-fire shot, a model was to be exercised to predict a test outcome, and that the predictions were to be provided prior to performing any full-up testing. Shot predictions were developed for the Bradley program using a relatively untested model called VAST (2). For multiple technical reasons, the model predictions (viz-a-viz the field results) were criticized. The reasons for the apparent disconnect between the predictions and the field results were manifold, but can be summarized, in the main, by the fact that the model was providing "lumped parameter" results based on key underlying, but unspecified, parameters. In fact, the net effect of the underlying phenomenology results in highly stochastic variations. However, up until this point, vulnerability studies were treated in a deterministic context.

In 1986, the Army began prosecuting the Abrams Live-Fire Program. To meet that challenge, a new vulnerability model was developed called SQuASH (3). At that time an abstract structure was developed to reason about the overall model logic (4). These notions were further refined in 1997 (5).

In 2003, major extensions were made to the logic (6); they included a] introducing official Universal- and Service-based tasks as the measures of mission effectiveness, b] an opposing force representation, c] chains of tasks sequencing, d] recursive representations by levels-of-war, e] top-down mission planning and f] a bottom-up mission execution processes. In 2005, Mr. Walter Hollis, the Deputy Under Secretary of the Army for Operations Research (DUSA-OR), requested that a demonstration be developed showing how the structure, called the Missions & Means Framework (MMF), might be employed to model and support operational testing. A computer simulation was developed (7) which represented a company of fighting vehicles, each modeled to a level of approximately 2000 components. This approach was based on the earlier live-fire efforts, and made possible a linkage from component damage to platform capability and then to platform utility. In this simulation actual vulnerability was not explicitly modeled; the intent of the exercise was to show how the damage to utility linkages played out and the connection between friendly and opposition forces.

In this past year, we were pleased to receive the support of the Army G-8 to make further progress in demonstrating what we believe is the highest resolution modeling of force-on-force (FoF) simulations to date.

Finally, we note that the MMF structure, though originally developed to couch ballistic interactions on military platforms, has expanded into a full description of military combat to include interactions of abstract nature. In so doing, the MMF is now seen as a general, collective ontology from the viewpoint of semantic categorization (8) (9).

Technical Discussion

1 - Background

Since the 1960s onset of modern Force-on-Force (FoF) modeling, simulations have been driven by task sequences defined by computer programmers. That was necessary since for many decades, military operators/warfighters had no standard task language. In the 1990s, official Joint and Service task lists were developed, establishing formal, doctrinally-linked semantics for the warfighter. This language construct enables evaluation of individual system contribution to collective tasks (the singular source for System-of-System conduct), mission performance and effectiveness.

The Missions and Means Framework (MMF) (Figure 2) provides an integrated procedure for deriving mission requirements in accordance with Joint Capabilities Integration and Development System (JCIDS) guidance and specifying them in a detailed and formalized structure that enables clear understanding by systems engineers and scientists in the research and development (R&D) and acquisition community as well as Test and Evaluation (T&E) professionals without compromising understanding by senior decision makers.

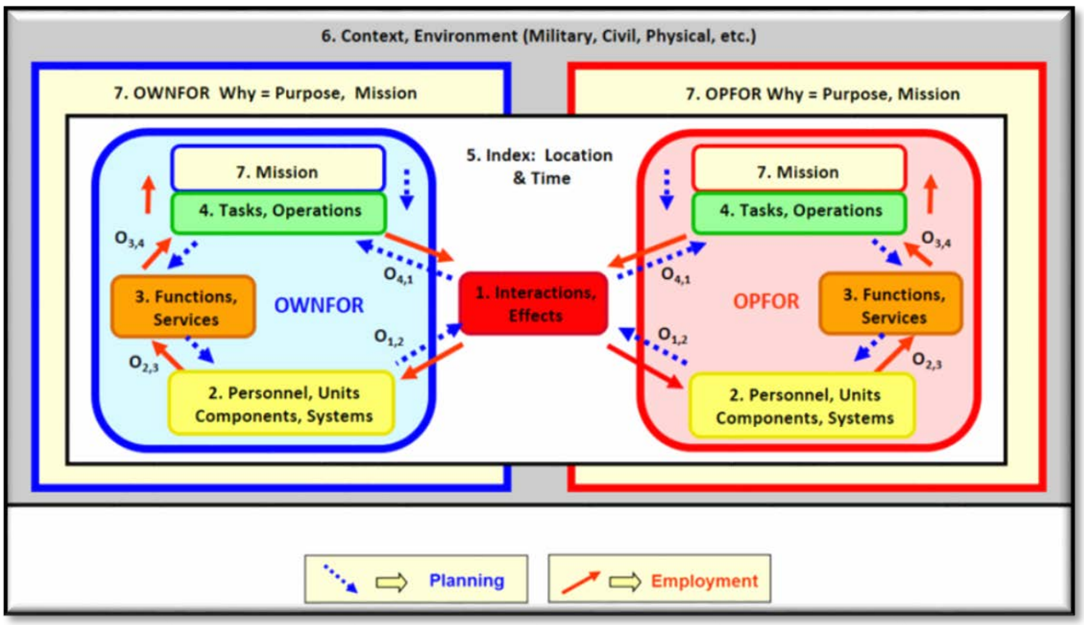


Figure 2 The MMF Conceptual Model

Figure 3 briefly illustrates the Army's process for developing operational requirements today. Military subject matter experts (SMEs) apply their knowledge of relevant Army doctrine and expertise in the Military Decision Making Process (MDMP) to analyze concepts and identify capability gaps related to the Army's ability to execute those concepts years in the future. The resulting analysis and operational requirement documents are written using doctrinally correct language for a target audience of senior military officers who must review and approve the requirements.

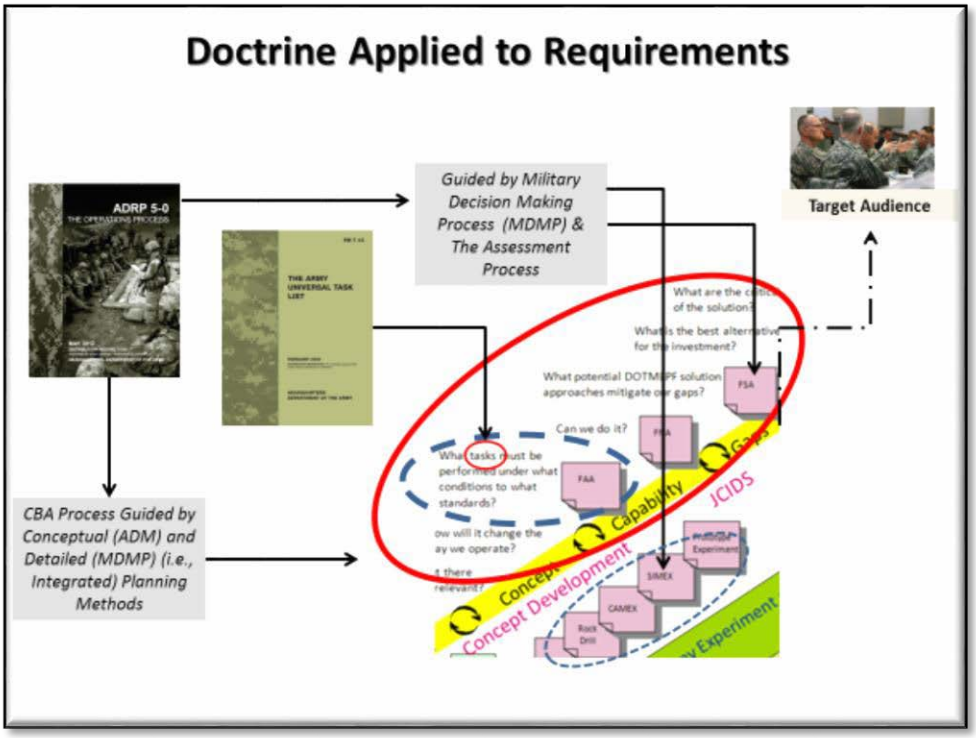


Figure 3 Developing Operational Requirements

Problems often begin within major acquisition programs when operational requirements, stated using this type of domain specific, doctrinal language, are found to be vague, confusing, and/or incomplete by the systems engineers who must translate them into system level requirements and ultimately into a detailed design. As shown in Figure 4, the current acquisition process leaves a considerable gap between the time operational requirements are first developed and the actual start of a formal acquisition program at Milestone B when the acquisition program baseline (APB) is established and development of detailed system requirements and specifications begins. This time gap contributes mightily to a persistent problem of exponential cost growth from the APB estimate as well as operational test failures and program cancellations.

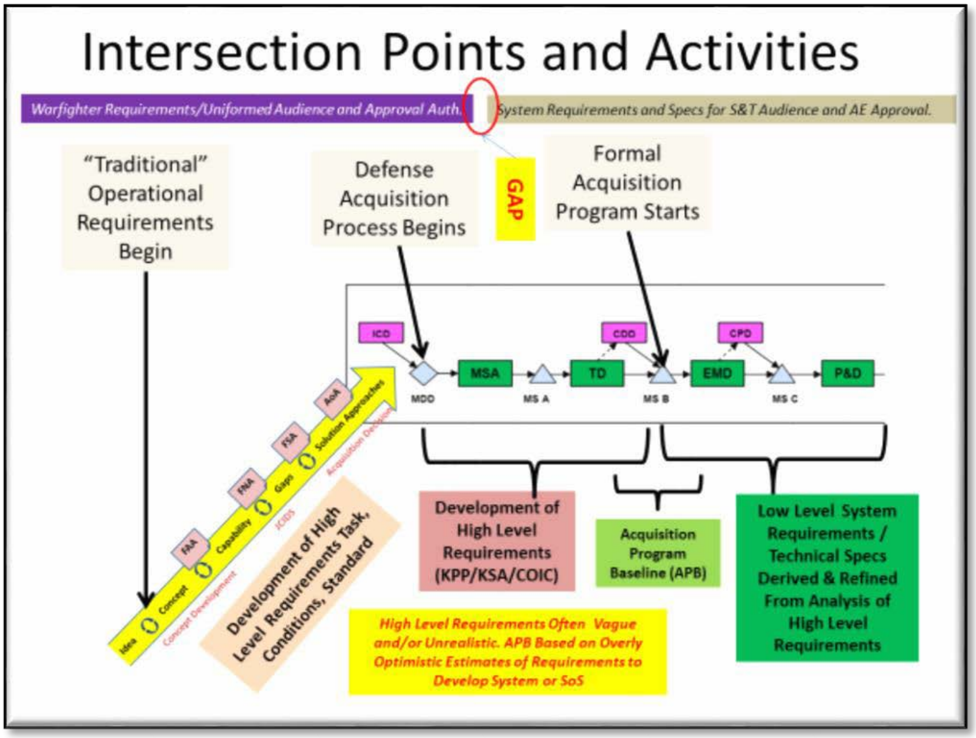


Figure 4 Acquisition Timeline and Requirements

Since the MMF was first developed and published over 13 years ago, we have learned much about how the framework can be applied to address the problems of developing and refining requirements using methods that can mitigate and overcome these problems. The italicized text boxes in Figure 5 below illustrate how application of the MMF levels to ask questions and store resulting answers can be applied to support Training and Doctrine Command's (TRADOC) capability based assessment (CBA) process (10) to generate operational requirements products required by the Joint Capabilities Integration Development System (JCIDS) (11). Immediately following, the italicized text boxes in Figure 6 illustrate how MMF levels can be applied by systems engineers after Milestone A to ask more detailed questions to develop detailed system requirements linked back to operational requirements and their source. Finally, Figure 7 illustrates how scalability of the MMF to analyze concepts at all levels of war may be applied to visualize, understand, and specify service and joint acquisition program inter-dependencies and inter-relationships.

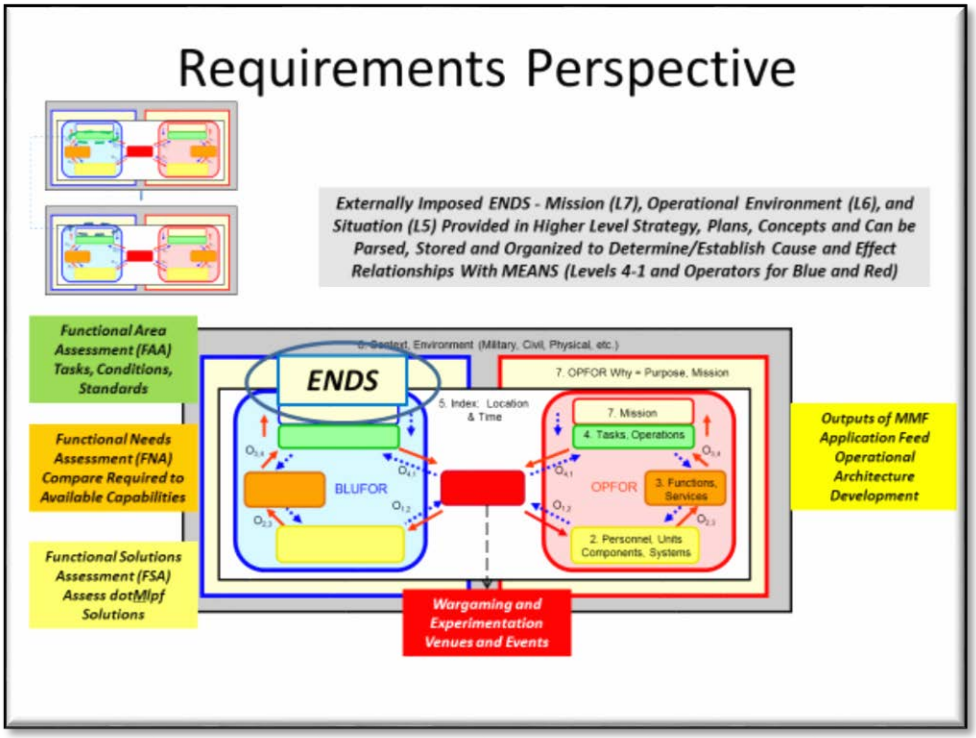


Figure 5 MMF Application to Requirements

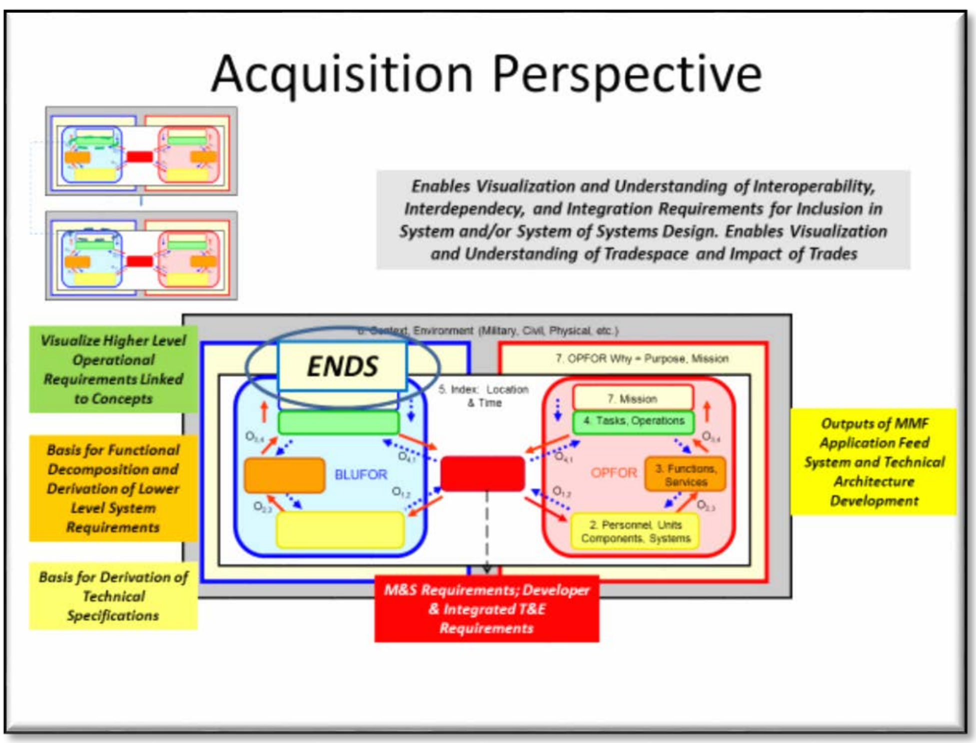


Figure 6 MMF Application to Acquisition

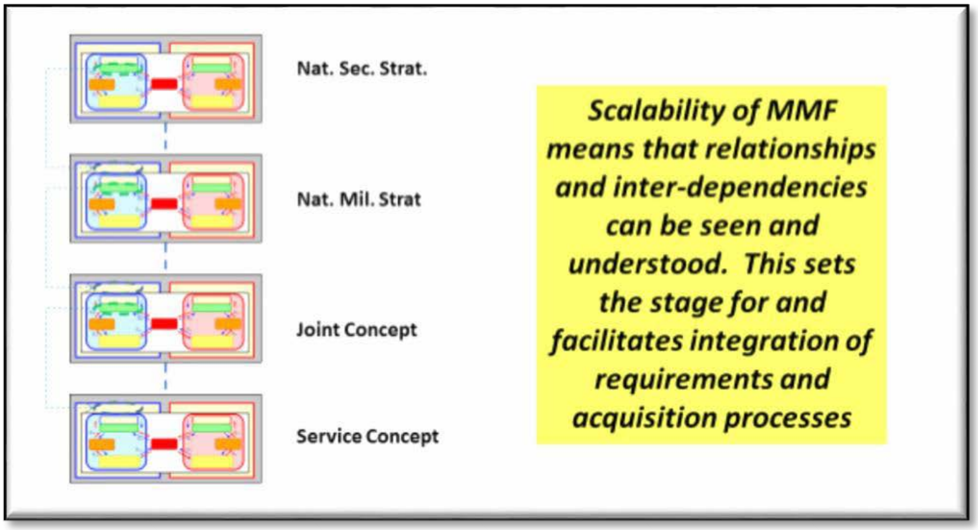


Figure 7 Leveraging MMF Scalability to Trace Requirements

Also since the 1960s, FoF models have employed combat entities with assigned, unchanging attributes. Interactions have focused on ballistic events, on pristine platforms, for kill/ no-kill outcomes. However, since the 1980s, platform models have existed to support detailed, mutable geometry so as to maintain a running status of component state space. This state space can be mapped to platform capabilities and then compared to task requirements per the formal task descriptors.

FoF models have also provided little to no value in assessing the value of information generated from various sources to mission required information. This is a critical shortfall in an era of irregular warfare and counter-terrorism where accurate and reliable information is essential and potential sources of information are exponentially greater. The MMF can be leveraged to determine mission essential information requirements and compare them to information generated by a variety of simulated sources.

2 - Mission Specification

2.1 Approach to mission specification to support T&E

When a system of interest/system under test is to be evaluated, mission specification is the process used to describe all elements of the supported mission to the level of detail required to:

1. Describe and/or reiterate the operational mode summary/mission profile (OMS/MP) describing how the Army envisions employment of the system of interest/system under test.

2. Enable understanding of what the larger SoS (an Armored Brigade Combat Team (ABCT), in this instance) is trying to accomplish and why within an operational context that includes:

a. A realistic and relevant operational environment that mirrors conditions specified in relevant operational requirements documents including operating and functional concepts.

b. A description of the events (fictional or otherwise) leading up to the current operational situation and mission.

c. A description of how the larger SoS' mission fits within the larger operational framework vertically (i.e. higher and lower echelons) and horizontally (i.e. supporting, supported and flanking organizations).

3. Describe the operations and tasks, with associated conditions and standards, to be performed as part of the Course of Action (COA)/Plan developed to accomplish the mission for a specific vignette.

4. Enable understanding of what capabilities are required to enable those operations and tasks.

5. Enable understanding of the combination of functions required to generate each required capability.

6. Describe the forces, resources and their associated organizational structure available to execute the COA – especially those forces which include the system(s) of interest in their organizational structure.

7. Describe the intended employment of available forces and resources through development of task organization; COA statements and sketches, execution matrices and other relevant products of the planning process.

8. Describe the sequence of planned interactions and their intended effects leading to accomplishment of the mission and capture in the form of a Time Ordered Event List (TOEL).

9. Describe the transformational logic applied to assess ability to perform tasks that are:

a. Performed by the system(s) of interest (e.g. AMPV Mortar Carrier) (Platform tasks).

b. Performed by the larger system(s) of systems (e.g. Mortar Platoon) (Collective tasks).

10. Describe the transformational logic applied to assess mission status and trace the status to lower level causes and effects.

2.2 Mission Overview.

Figure 8 below is derived from Version 1.1 of the draft AMPVOMS/MP (12). It described the anticipated mix of missions the ABCT will encounter in periods of peace, crisis, national conflict, and war along with the anticipated conditions the AMPV will operate in as part of the larger force.

Mission Overview

(Derived from AMPV Operational Mode Summary)

- **Major Combat Operations. (72 hours) AMPV equipped Armored BCT (ABCT)**
  - Conducts extended tactical movement to Assembly Area; Deliberate attack to seize an Objective followed by an Exploitation and then conducts a deliberate attack to capture final objective in a complex urban area.
- **Irregular Warfare Operations. (7 days) AMPV equipped Armored BCT (ABCT)**
  - Provides route and area security within Area of Operations (AO) to prevent remnants of enemy conventional forces and irregular forces from disrupting and threatening coalition and legitimate government operations.
- **Peace Operations (170 days) AMPV equipped Armored BCT (ABCT)**
  - Maintain stability and security within assigned AO, including complex urban area. Be prepared to support and command and control simultaneous reconstruction efforts

Major Combat Operations	Offensive	Defensive	Stability	Total (hrs)	Total (days)
Calendar Time (hrs)	49	15	8	72	3
Percentage	68.1%	20.8%	11.1%	-	-
Irregular Warfare	**Offensive**	**Defensive**	**Stability**	**Total (hrs)**	**Total (days)**
Calendar Time (hrs)	48	44	76	168	7
Percentage	28.6%	26.2%	45.2%	-	-
Peace Operations	**Offensive**	**Defensive**	**Stability**	**Total (hrs)**	**Total (days)**
Calendar Time (hrs)	396	796	2888	4080	170
Percentage	9.7%	19.5%	70.8%	-	-
Cumulative Total	**Offensive**	**Defensive**	**Stability**	**Total (hrs)**	**Total (days)**
Calendar Time (hrs)	529.5	919	2871.5	4320	180
Percentage	12.3%	21.3%	66.4%	-	-

Figure 8 Mission Overview

2.3 Operational Context.

The operational context for this study effort was derived from the unclassified Multi-Level Scenario Module 2: IX Corps (TRAC-F-TR-08-063). (13) Henceforth in this report the scenario will be referred to as MLS 2.0.

2.3.1 Why pick this scenario?

The study team needed a realistic operational scenario to provide the analytical space required to adequately assess AMPV contributions to the ABCT across the full range of the OMS/MP. MLS 2.0 is a unique, Defense Planning Scenario/Multi-Service Force Deployment (DPS/MSFD)-like, major combat operation (MCO) with a swiftly defeat characteristic. With an unlimited shelf life, the MLS was designed to serve as the foundation for longer term analysis and studies with the flexibility to be either classified or unclassified depending on the user's needs. MLS 2.0 was designed to aid the development of future system requirements and can support acquisition specification when classified.

2.4 Scenario Overview.

Given our necessary focus at the lower tactical level, we selectively applied relevant portions of the scenario to illustrate the relationship of the AMPV-equipped ABCT to the larger operational force from Division all the way up to Combined Joint Task Force (CJTF) at the Strategic Theater level of war. We then extended the scenario through the development of operational vignettes from the ABCT all the way down to the individual platform/squad/section level to enable assessment of platform (e.g. AMPV variants, M1, M2 and M3) ability to perform assigned tasks given dynamically changing platform geometry.

2.4.1 Illustration of JFC to ABCT relationships.

Figure 9 illustrates the hierarchical, task-based relationship from the Strategic National (SN) level of war at which the Secretary of Defense (SECDEF) and National Security Council (NSC) direct employment of military forces all the way down to the Tactical level of war at which the ABCT operates to achieve effects that are nested from Division to Corps to Joint Force Land Component (Operational level of war) and all the way to the Combined Joint Task Force (CJTF) at the Strategic Theater level. We will refer to this hierarchy throughout the scenario overview to orient the reader on the relationship of the portion of the scenario being described to the all-important AMPV-equipped ABCT.

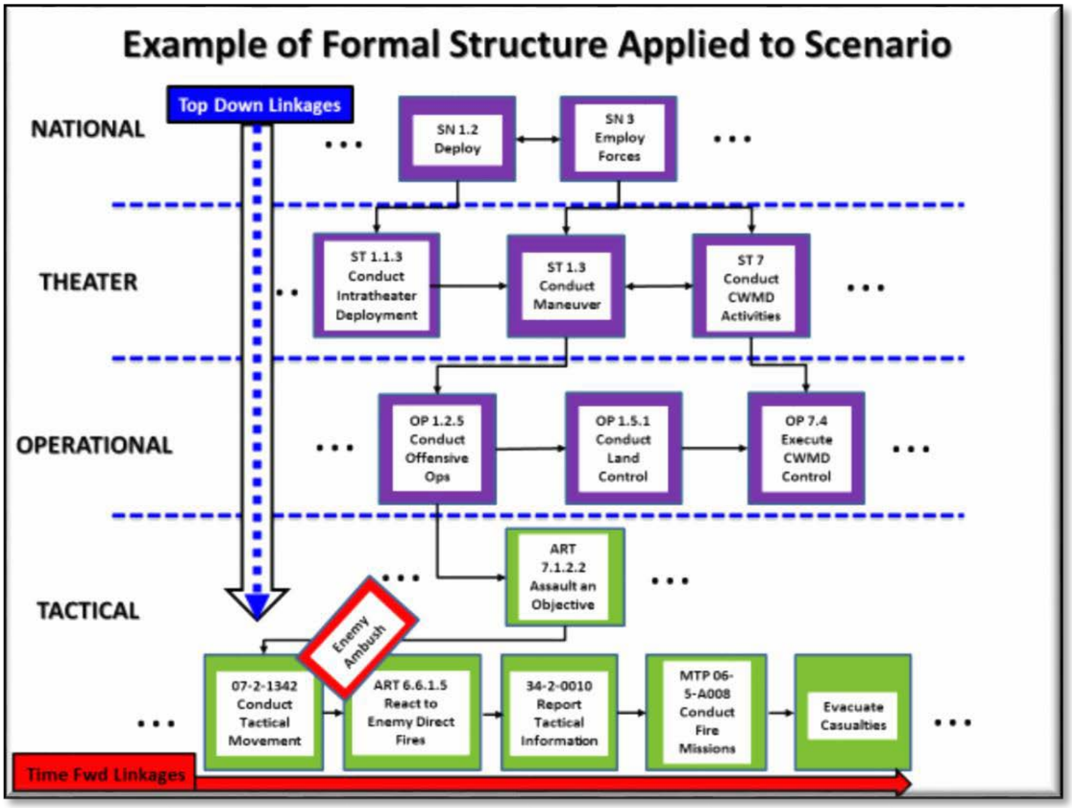


Figure 9 Illustration of Task-based Hierarchy

2.4.2 Theater Force Structure.

CJTF Freedom occupies the top rung in the military force structure for the operation depicted in this scenario. Figure 10 illustrates the organization of forces mapped to the task-based hierarchy depicted in the previous figure.

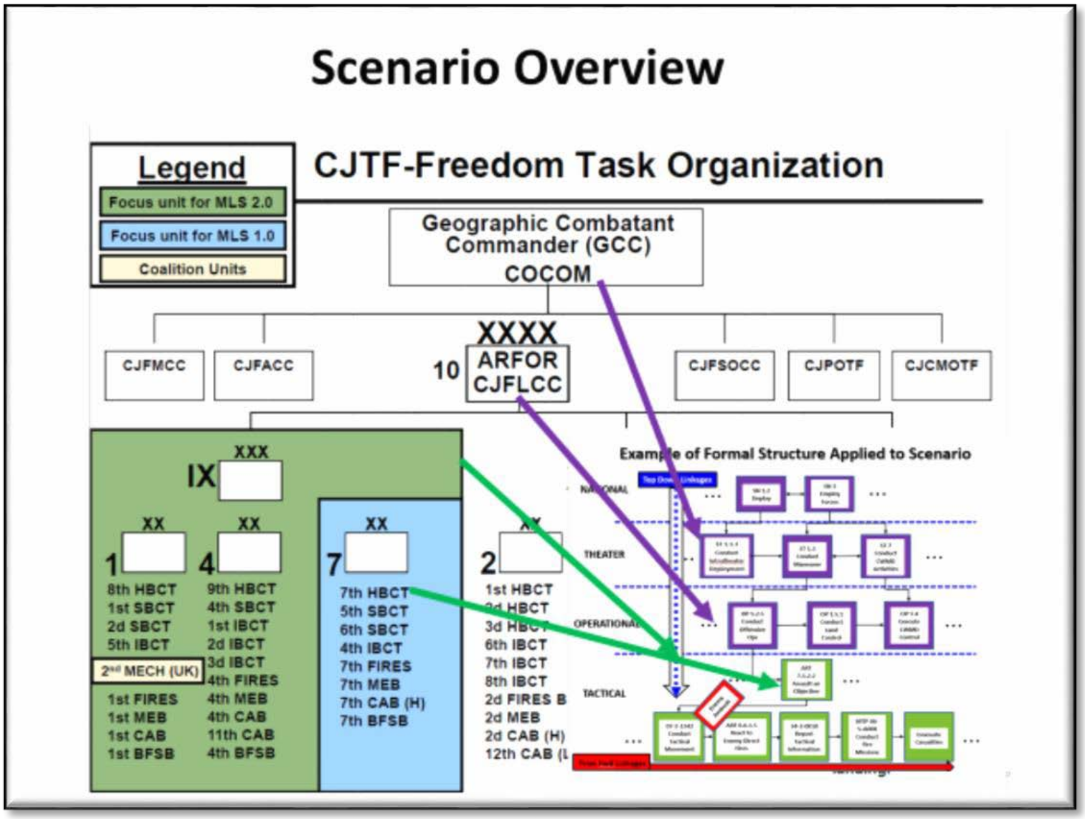


Figure 10 CJTF Freedom Organization for Combat

2.4.3 CJTF Freedom Mission.

Figures 11 and 12 illustrate the mission and Concept of Operations (CONOPs) for CJTF Freedom. (13) para. 5-6. The CONOPs includes Task and Purpose statements for all CJTF Freedom's functional (e.g. Land, Air, Maritime, etc.) commands. Note that we have circled those for the Combined Joint Force Land Component Command (CJFLCC).

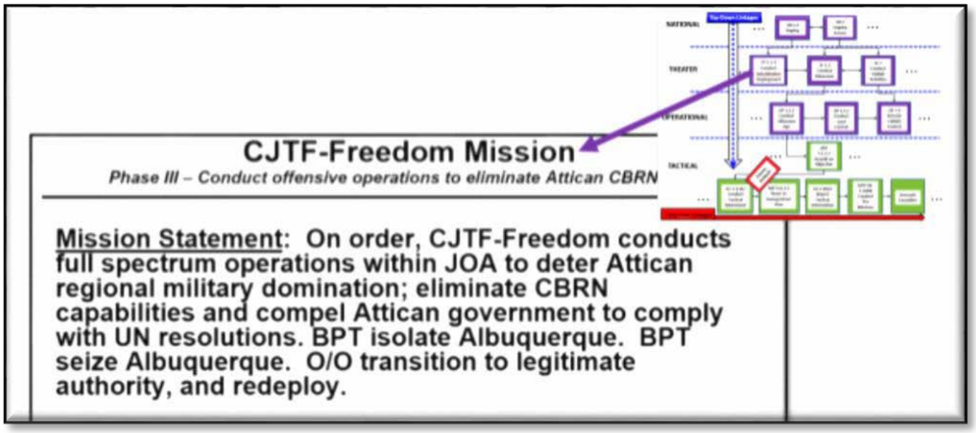


Figure 11 CJTF Freedom Mission

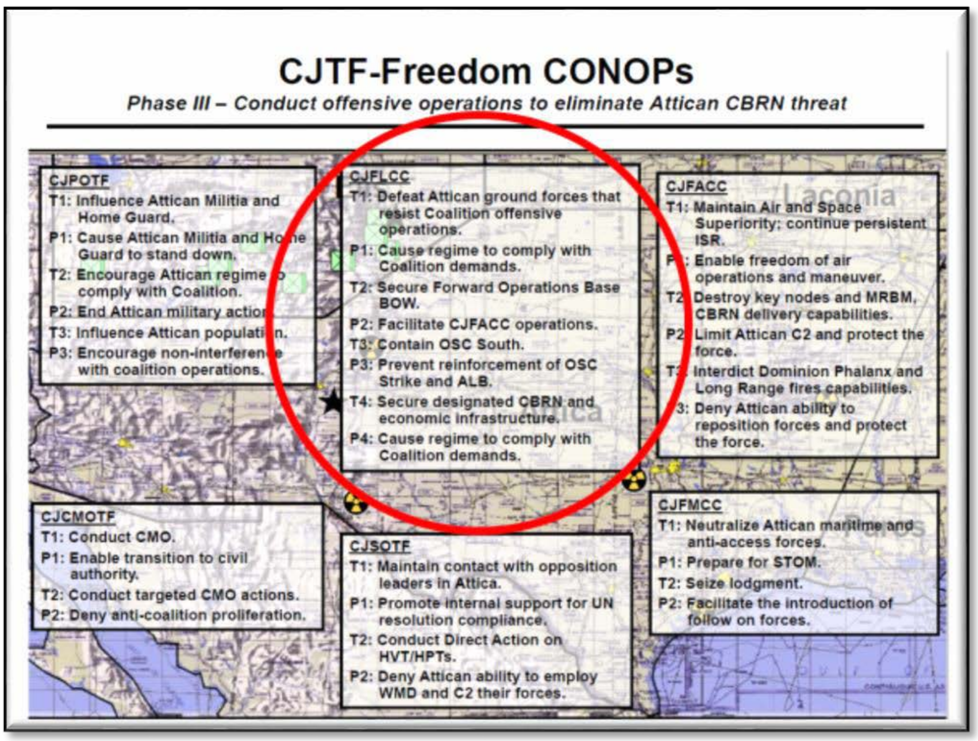


Figure 12 CJTF Concept of Ops and Tasks for CJFLCC

2.4.4. Combined Joint Forces Land Component Command (CJFLCC) Mission and CONOPs.

Figure 13 shows the mission of the CJFLCC, derived from the specified tasks and associated purposes provided in the CJTF CONOPs in Figure 12. The CONOPS for CJFLCC (Figure 14) includes specified task and purpose statements for subordinate Corps-level commands at the upper tactical/lower operational level of war. For purposes of the MLS 2.0 scenario, 10^{th} (US) Army, a service component command, is dual-hatted as the Combined Joint Force Land Component Command (CJFLCC) with control of U.S. Marine Corps forces in addition to U.S. Army forces.

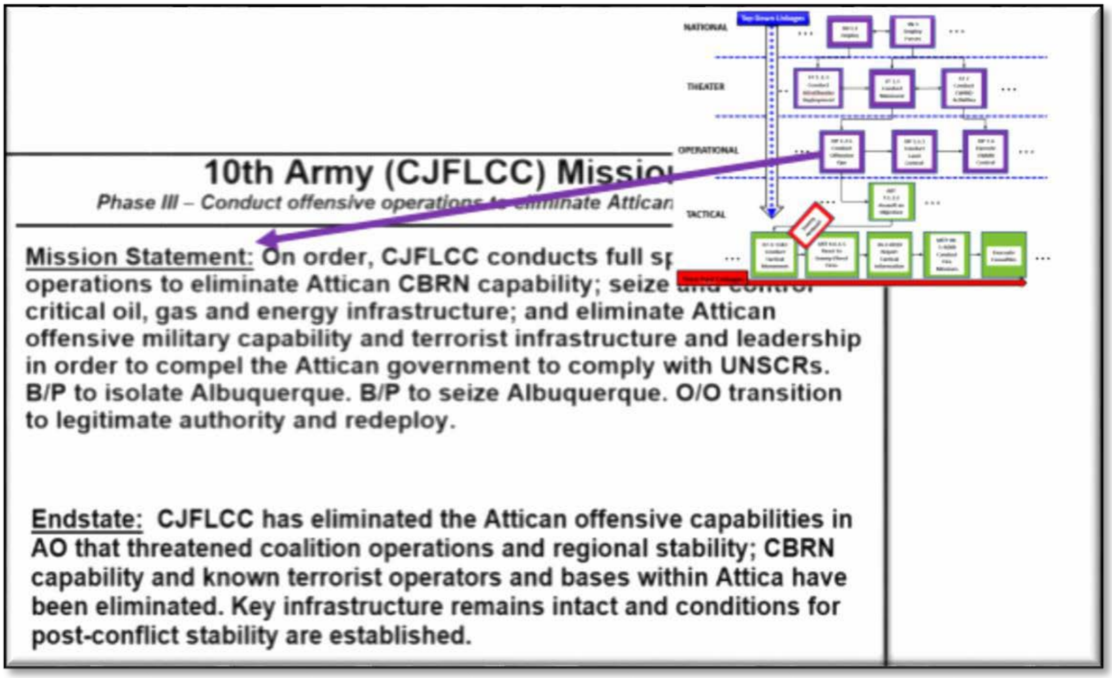


Figure 13 CJFLCC Mission and End State

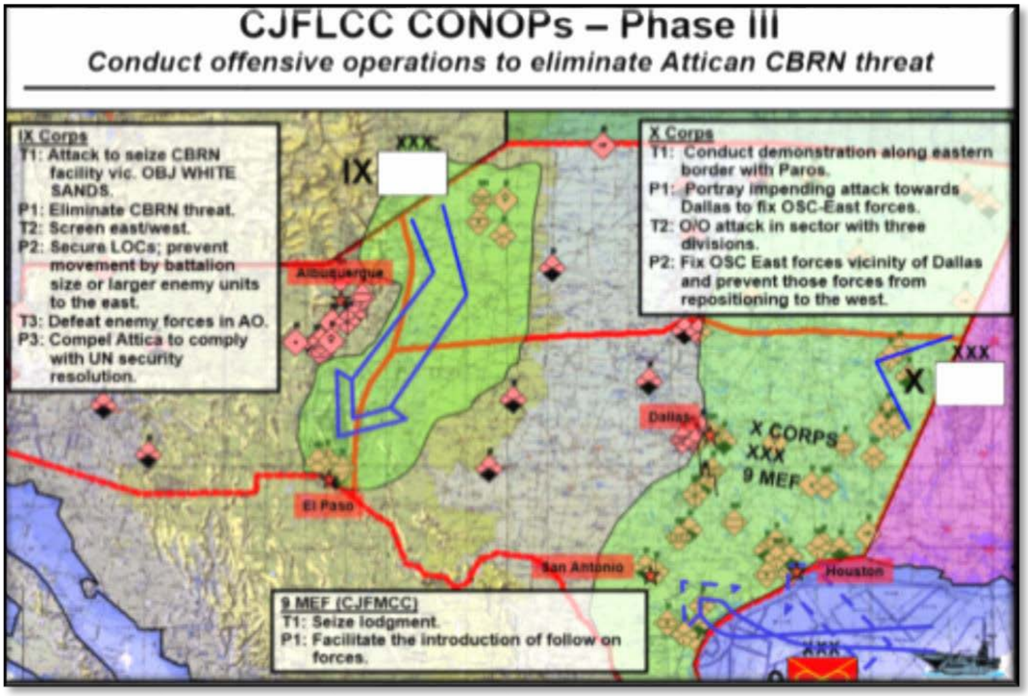


Figure 14 CJFLCC CONOPS

2.4.5 IX Corps CONOPs.

Our hierarchical drill-down continues by narrowing the focus to one of the three subordinate Corps-level organizations in the CJFLCC, the IX Corps. Figure 15 illustrates the IX Corps CONOPs, focusing in on the planned attack by one of the three subordinate Divisions, the 7th Division, in the IX Corps.

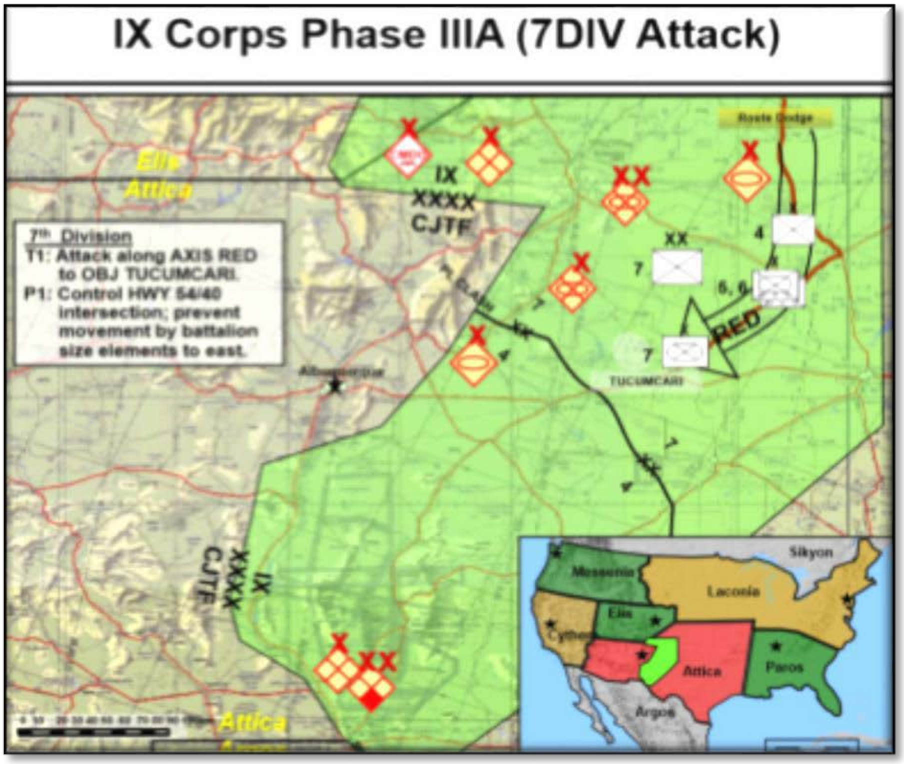


Figure 15 IX Corps CONOPs

2.4.6 7TH Infantry Division Attack.

Figure 16 shows the IX Corps task organization with a total of three Divisions and multiple Corps enablers (e.g., MP Brigade, Air Defense Brigade, etc.). We focused on one of the three Divisions, the 7th Infantry Division (7th ID), because it includes an ABCT and its assigned mission in MLS 2.0 Scenario encompasses the OMS/MP for the AMPV. In addition, the Peace Operations portion of the mission (to be conducted in the urban area of Tucumcari), provides analytical space for potential follow on studies and analyses designed to compare requirements for information to discoverable information sources. Figure 17 shows the 7th ID CONOPs including the 7th ABCT's attack to secure Tucumcari.

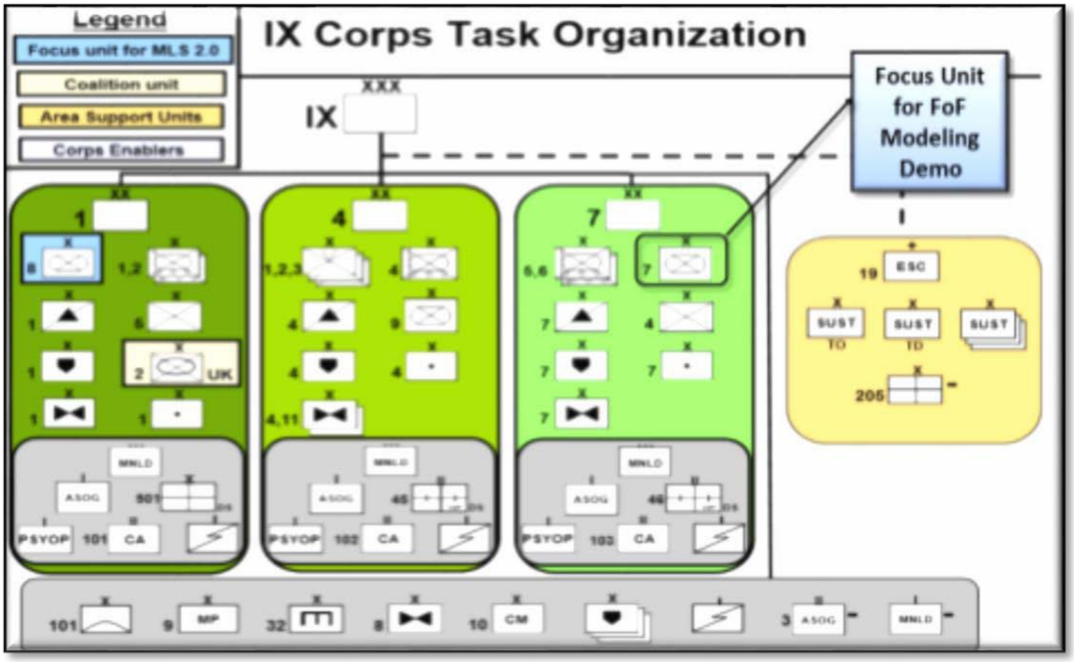


Figure 16 IX Corps Task Organization with 7ID Task Organization

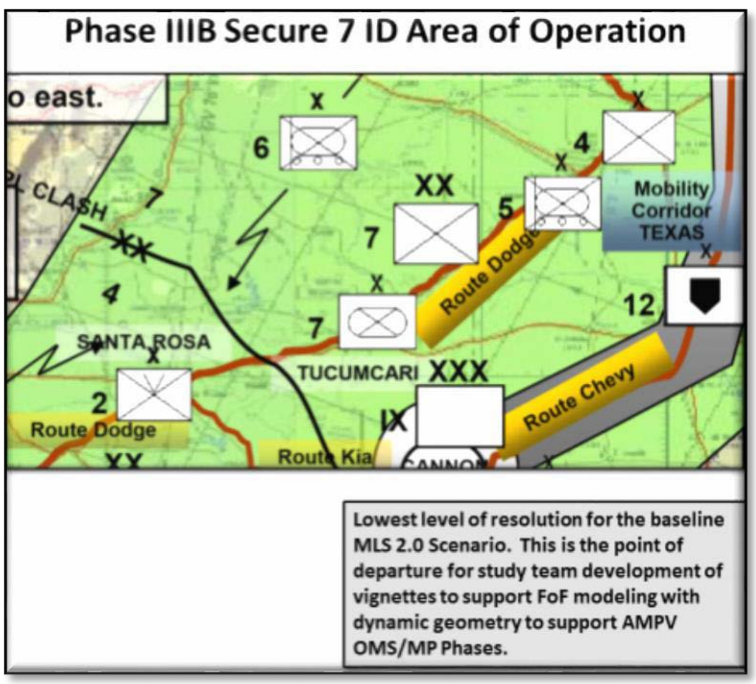


Figure 17 7th ID CONOPs

2.4.7 7th ABCT Deliberate Attack and Exploitation.

This is one of the specified mission types called for in the AMPV OMS/MP. We chose to focus on this mission for purposes of the demonstration due to the increased likelihood of incurring kinetic effects and relative ease of linking the effects of kinetic and reliability interactions to mission impact. Figure 18 illustrates the 7th ABCT CONOPS for this mission while also showing where it fits in the Scenario's Mission – Task Hierarchy.

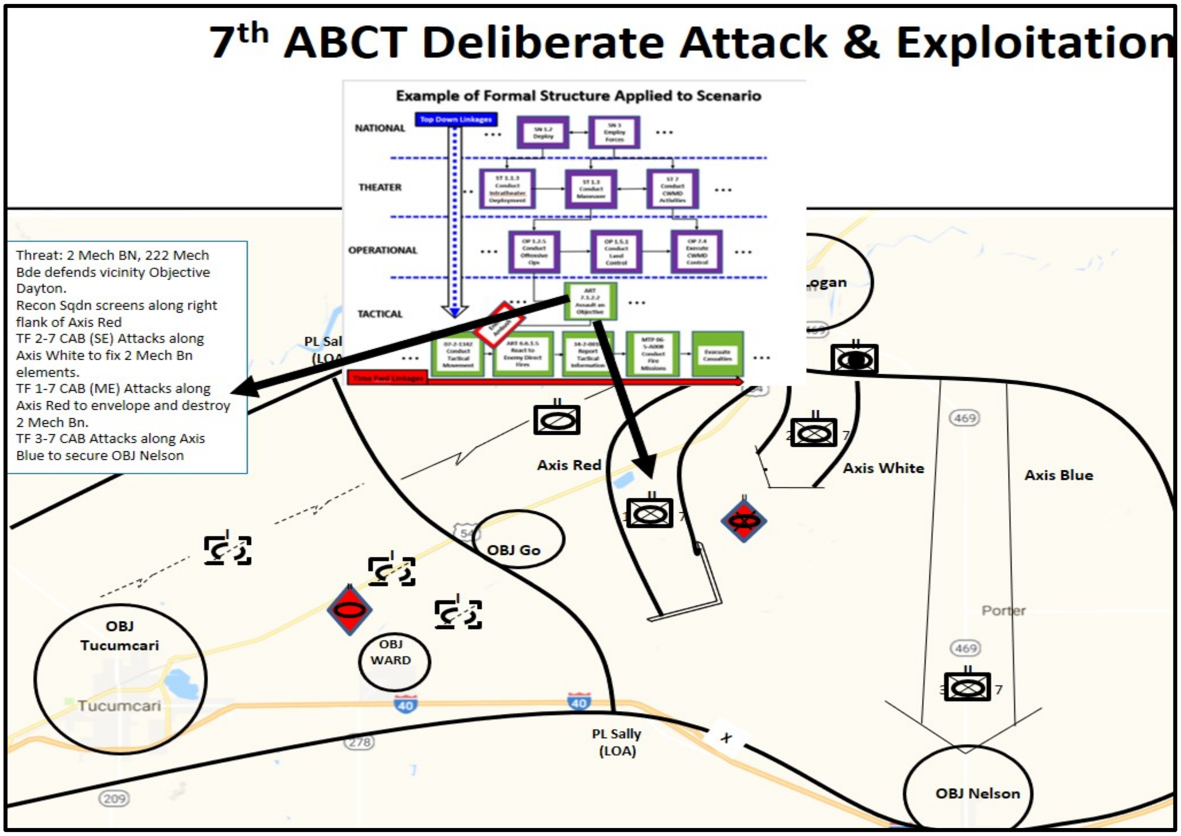


Figure 18 7th ABCT CONOPs for Deliberate Attack and Exploitation

2.5 MMF Application.

Up to this point, the mission specification process generated products from existing operational requirements and scenario documentation along with further analysis using the MDMP to extend standard scenarios. We extended the standard scenario involved by applying the MDMP to: Analyze the task and purpose assigned to the 7th ABCT by the 7th Division for Phase IIIB Secure; Develop a restated mission; and then, generate a COA to accomplish it. This process was replicated for the 1st Combined Arms Battalion (CAB) (CAB1), 7th ABCT and Company/Team A of CAB1 along with the Mortar Platoon and Medical Platoon for CAB1 (See Appendix A, MDMP products). The purpose of this top down mission analysis and planning process was to generate the operational vignettes needed to describe missions and COAs for lower tactical level organizations down to the level required to incorporate dynamic geometry for systems of interest.

2.5.1 Kinetic Formal MMF Process Flow Diagram.

Figure 19 illustrates the Kinetic Formal MMF Process Flow Diagram which was used to guide the application of the MMF and population of the appropriate MMF levels and operators with information derived from the MDMP process along with source documents for authoritative names and descriptions of tasks, organizations and systems.

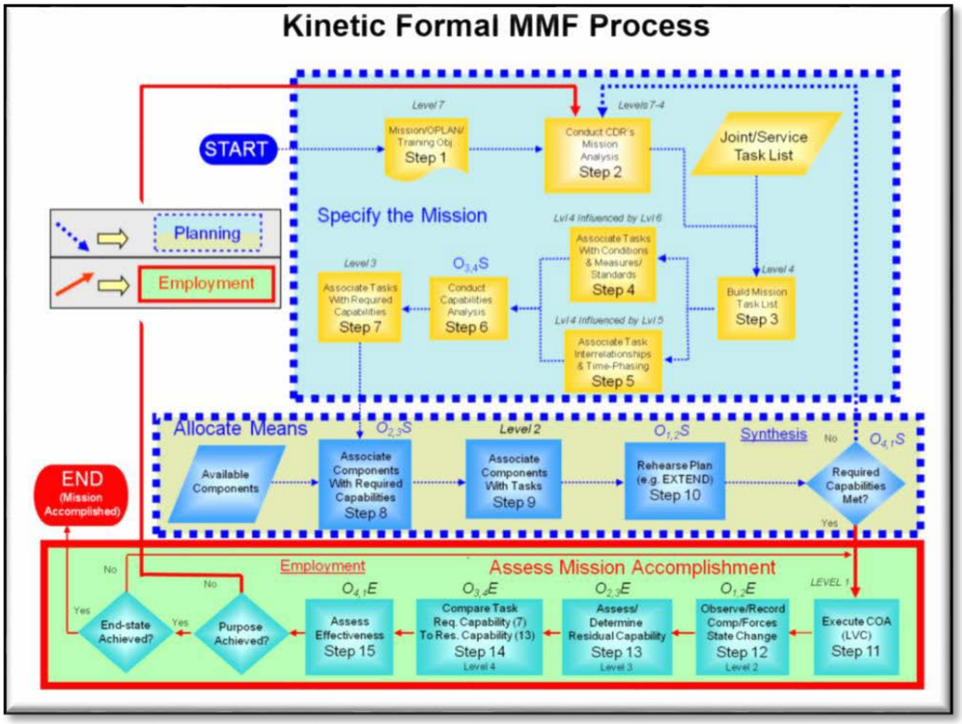


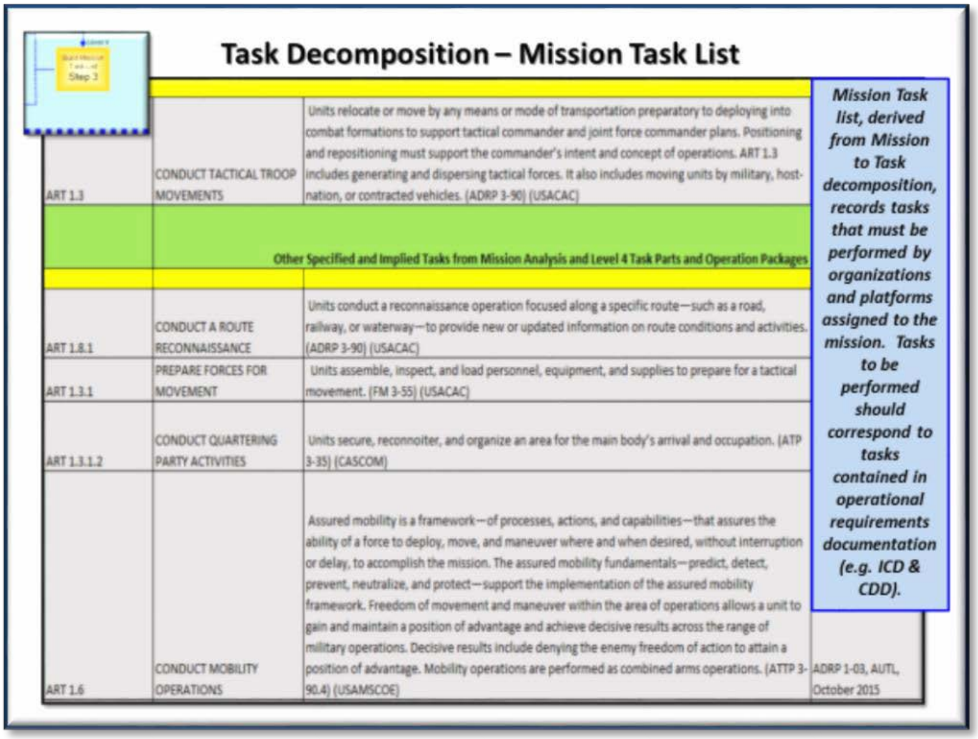
Figure 19 The Kinetic Formal MMF Process Flow Diagram

2.5.2 Mission Task list.

Figure 20 provides an example of a portion of the mission task list for the Deliberate Attack and Exploitation mission. The mission task list is an output of Step 3 of the Kinetic Formal MMF Process Flow Diagram (Figure 19). Steps 1 and 2 of the process were performed during application of the MDMP to develop the operational vignettes as described in para. 2.5 above. Step 3 required conversion of specified, implied and essential tasks identified during Step 2 (Conduct Commander's Mission Analysis) into the formal language contained in the authoritative task lists (ATL) below:

- Army Doctrinal Reference Publication (ADRP) 1-03, The Army Universal Task List (AUTL) (14)
- Training Circular (TC) 3-90.6, Brigade Combat Team Collective Task Publication (15)
- Training Circular (TC) 3-90.5, Combined Arms Battalion Collective Task Publication (16)
- Training Circular (TC) 3-90.1, Armor and Rifle Company Team Collective Task Publication (18)
- Training Circular (TC) 3-21.8, Infantry Rifle and Mechanized Platoon Collective Task Publication (20)
- Training Circular (TC) 3-20.15, Tank Platoon Collective Task Publication (22)
- Training Circular (TC) 3-20.98, Reconnaissance Platoon Collective Task Publication (24)
- Training Circular (TC) 3-21.90, Mortar Platoon Collective Task Publication (26)

Complete mission task lists and other MMF products generated for this demonstration may be requested from the authors.



ART 1.3	CONDUCT TACTICAL TROOP MOVEMENTS	Units relocate or move by any means or mode of transportation preparatory to deploying into combat formations to support tactical commander and joint force commander plans. Positioning and repositioning must support the commander's intent and concept of operations. ART 1.3 includes generating and dispersing tactical forces. It also includes moving units by military, host-nation, or contracted vehicles. (ADRP 3-90) (USACAC)	
	Other Specified and Implied Tasks from Mission Analysis and Level 4 Task Parts and Operation Packages		
ART 1.8.1	CONDUCT A ROUTE RECONNAISSANCE	Units conduct a reconnaissance operation focused along a specific route—such as a road, railway, or waterway—to provide new or updated information on route conditions and activities. (ADRP 3-90) (USACAC)	
ART 1.3.1	PREPARE FORCES FOR MOVEMENT	Units assemble, inspect, and load personnel, equipment, and supplies to prepare for a tactical movement. (FM 3-55) (USACAC)	
ART 1.3.1.2	CONDUCT QUARTERING PARTY ACTIVITIES	Units secure, reconnoiter, and organize an area for the main body's arrival and occupation. (ATP 3-35) (CASCOM)	
ART 1.6	CONDUCT MOBILITY OPERATIONS	Assured mobility is a framework—of processes, actions, and capabilities—that assures the ability of a force to deploy, move, and maneuver where and when desired, without interruption or delay, to accomplish the mission. The assured mobility fundamentals—predict, detect, prevent, neutralize, and protect—support the implementation of the assured mobility framework. Freedom of movement and maneuver within the area of operations allows a unit to gain and maintain a position of advantage and achieve decisive results across the range of military operations. Decisive results include denying the enemy freedom of action to attain a position of advantage. Mobility operations are performed as combined arms operations. (ATTP 3-90.4) (USAMSCOE)	ADRP 1-03, AUTL, October 2015

Figure 20 Mission Task List Example

2.5.3 Mission thread development.

Figure 21 provides an example of the task-based mission thread developed for the Deliberate Attack and Exploitation phase of the AMPV OMS/MP. Steps 4 and 5 of the Kinetic Formal MMF Process Flow Diagram were performed to generate the mission thread which provides the structure needed to conduct the bottom up assessment of performance and mission effectiveness as the COA with associated tasks is being executed. If generating a COA to accomplish a mission is part of the military art, then wargaming that same COA is part of military science. The controlled "Action-Reaction-Counter Action" nature of wargaming in conjunction with detailed recording produces a clear understanding of the timing and condition-based dependencies that establish the sequence of "desired effects" needed to achieve the end state for a particular mission along with the set of "interactions" deemed most likely to produce those desired effects. With that information as a starting point we used military expertise and logic to select the tasks needed to generate likely interactions and associate the set of conditions and standards most descriptive of the minimum levels of performance and effectiveness for resulting interactions to achieve desired effects. Analysis of those tasks, conditions and standards yields an understanding of capabilities required to achieve desired effects (the purpose of the task); and the combination of functions required to generate the performance (e.g. speed, accuracy, lethality, etc.) needed to enable the capability.

A football team for example, plans to execute a certain passing play to score the winning touchdown against their main rival. They require the capability to legally move the football over 40 yards in cold weather on a muddy field against a very tough defense. To achieve that capability, they must be able to block defensive rushers for at least 4 seconds; run their passing routes in 4 seconds or less; throw the football in 4 seconds or less to a point within 2 feet of the receiver's hands while he is running full speed; catch a football thrown within 2 feet of the hands; hold on to the ball after being hit by a defensive player; and keep both feet inbounds.

Armed with an understanding of required capability and functions, we can assign systems (***e.g. particular blockers, receivers and quarterback***) that possess the required functions needed to generate the capability to successfully execute the task and accomplish its purpose (***i.e. block, run, throw, catch and score a touchdown without penalty***). The mission thread records how the COA is expected to play out if events proceed per plan. It is task-based because we use tasks from the mission task list to describe actions being taken by systems (platform tasks) and systems of systems (collective tasks). Just as the planning process provides for development of potential branches and sequels at key points, one can generate mission threads for each branch or sequel that is planned with its own COA.

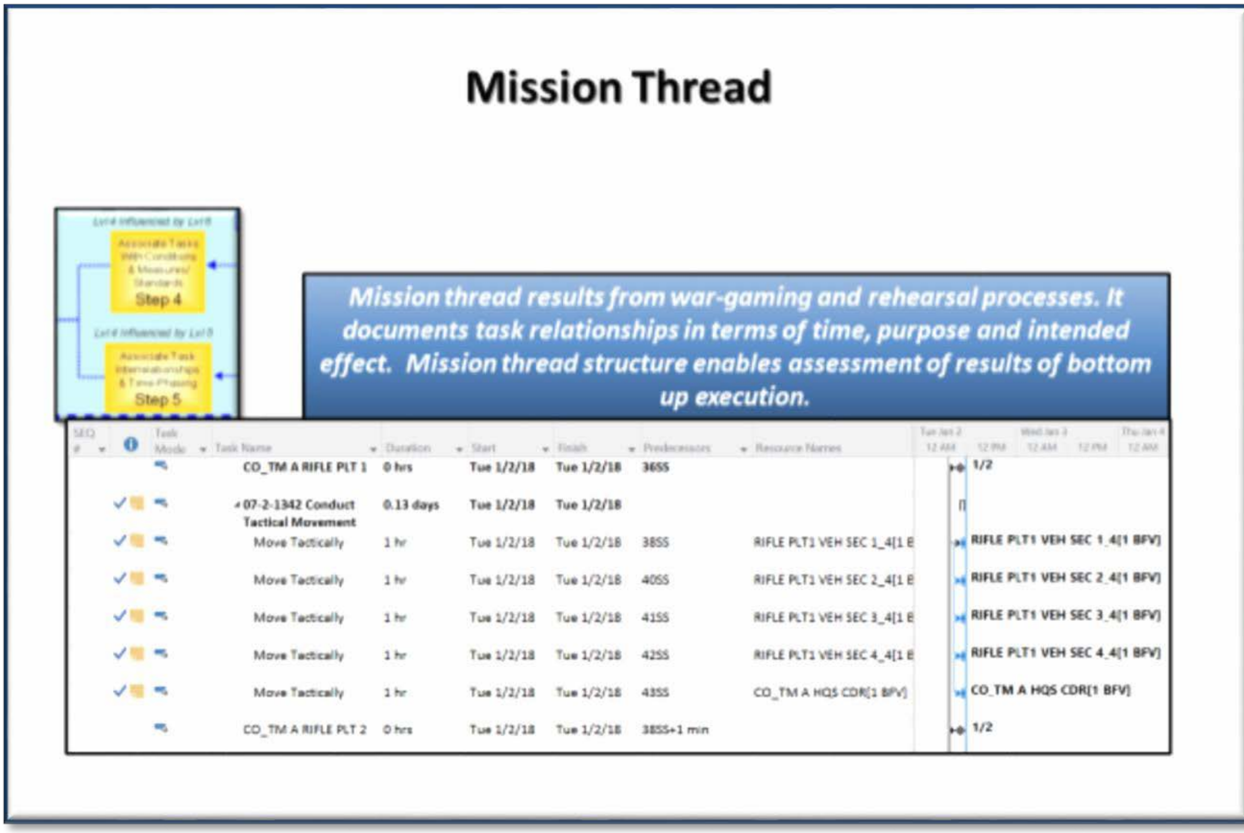


Figure 21 Mission Thread

2.5.4 Platform and Collective Tasks.

In football terms one can think of a collective task like a play to be executed by the offensive team. For the play to be successful, each player on the offense is given an assignment such as pass blocking, route running, faking a run, etc. One can think of these individual assignments as platform tasks. A mechanized infantry platoon conducting an attack while mounted on their M2 Bradley Fighting Vehicles (BFV) is conducting a collective (platoon-level) task. Each BFV in the platoon must perform a series of tasks (e.g. move tactically, identify targets, engage targets, etc.) to make the platoon level collective task successful. These are examples of platform tasks. Why is the difference between platform and collective tasks important to understand?

Platform tasks require functions that are directly enabled by the platform and its components to generate the capability needed to accomplish the purpose for each task. This is the point at which dynamic geometry becomes very important. Changes in the state of the platform and its components necessarily change the state of functions and capabilities available to perform assigned tasks and achieve the required capability.

Collective tasks are composed of lower level tasks, including platform tasks and tasks that are potentially performed by more than one platform. Successful performance of a collective task and accomplishment of its assigned purpose depends on the outcome of some set of the lower level tasks that comprise it. The pass play for our football team for example may be successful if one or two blockers fall short and only hold their blocks for 2 seconds but an inaccurate throw by the quarterback or dropped pass by the receiver dooms the play.

This relationship between collective and lower level tasks forms the basis for developing the logic to model the execution and assessment of collective tasks assigned to complex organizations/Systems of Systems. The relationship holds true for collective tasks comprised of sets of platform tasks and for collective tasks comprised of lower level collective tasks and combinations of platform and lower level collective tasks. An attack conducted by a mechanized infantry company/team for example, could be comprised of lower level collective tasks assigned to its two mechanized platoons and its one tank platoon as well as mission command-related tasks assigned to the commander or First Sergeant's vehicle (platform tasks). In the football analogy one could think of the offensive team as the company and the sets of offensive linemen and receivers as individual platoons with the quarterback performing mission command.

These relationships are captured in the mission thread when it is properly constructed. Specification of task relationships in the mission thread enables the analyst to visualize and understand the impact of degraded states and task failures caused by discrete events.

It was vital to determine and capture those task relationships. Later in this report we will illustrate how task relationships were captured for the Deliberate Attack and Exploitation mission and ultimately incorporated in the model.

2.5.5 Specifying Required Capabilities for Platform Tasks.

The mission specification team determined required capabilities and functions as part of the war gaming, or COA analysis portion of the MDMP. This step is necessary for this study because we wanted to integrate dynamically changing component and platform states for the systems of interest.

The system specification team provided the mission specification team with the complete set of 19 capability state descriptors that could potentially result from kinetic interactions or reliability failures simulated by the system vulnerability model (MUVES).

We needed them to specify what subset of capability state descriptors would be required to successfully perform each task assigned to one or more of the systems of interest. This specification was captured in a Capability to Task matrix and later incorporated in the model logic. The capability to task matrix at Figure

22 illustrates the format used to specify capabilities required of the platforms included as systems of interest for this study: AMPV (all variants); M2 BFV; M3 Cavalry Fighting Vehicle (CFV); and the M1 Tank.

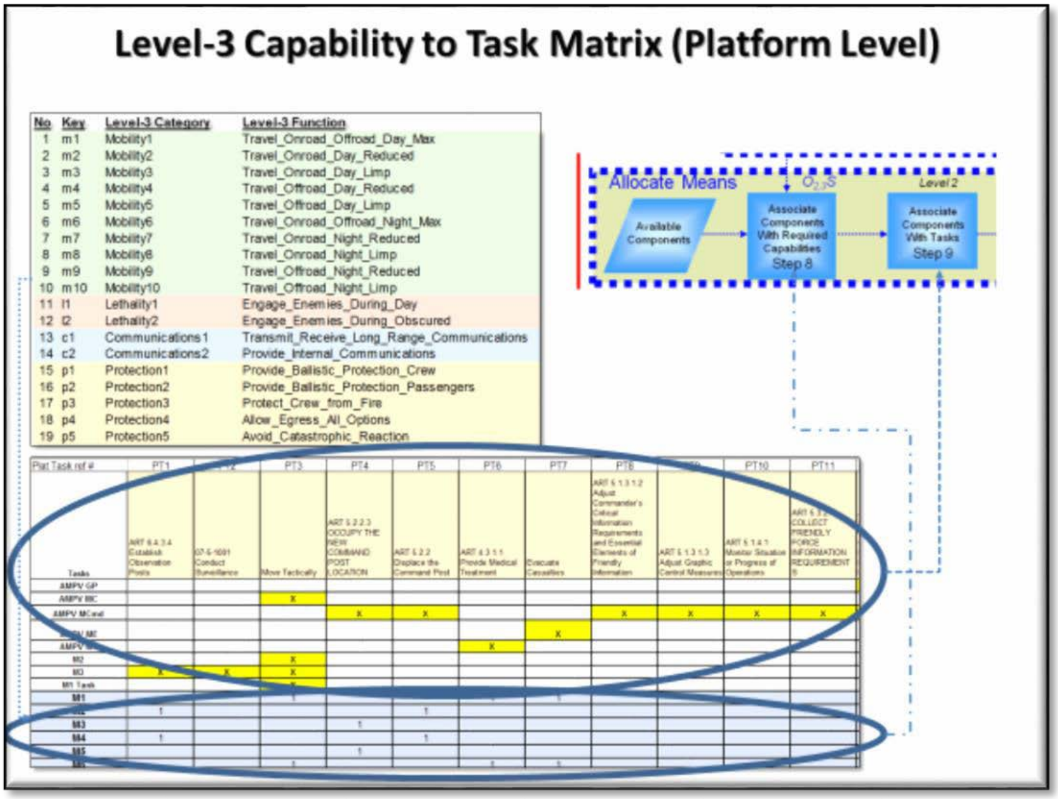


Figure 22 Capability to Task Matrix

2.5.6 Specifying Required Capabilities for Collective Tasks.

A tactical organization (e.g., platoon, company, battalion, etc.) requires subordinate and supporting organizations to perform assigned tasks that together comprise a collective task for that organization. When a mechanized CO/TM, for example, conducts a tactical movement task to reach a new position, each of its subordinate platoons and sections must perform tasks that enable successful movement of the company as a collective whole. The CO/TM commander assesses situational conditions (e.g., threat, weather, terrain, follow-on tasks, etc.) impacting movement and establishes standards for success. The resulting "task, conditions, standard" statement might be:

CO/TM A conducts a tactical movement (task # 07-2-1342) to reach Attack Position Echo. Weather and road conditions are good, but be prepared for the threat of Improvised Explosive Devices (IEDs) and ambush by small teams of bypassed enemy forces along the route. Success will be arrival at the Attack Position with at least two thirds of our combat power by no later than H hour plus 30 minutes.

For the collective task of "Conduct tactical movement" in this situation, The CO/TM requires the capability to complete its tactical movement to the Attack Position with at least 2 of its 3 platoons by H + 30 minutes. Achieving this capability requires that at least 2 of the 3 platoons of CO/TM A successfully perform their platoon-level collective tasks of "Conduct tactical maneuver". Figures 23 and 24 illustrate the format used to communicate required capabilities for collective tasks to the simulation developer to enable selected collective tasks to be simulated and assessed in the FoF model.

Task assessment during actual mission execution entails establishment of measurable and observable metrics (Measures of Performance (MoP) and Measures of Effectiveness (MoE)); observation of task performance; measurement of key performance variables; measurement of key effectiveness variables; and comparison of measured results to established standards.

It is often not possible to realistically observe and measure performance or effectiveness for discrete tasks in a simulation environment. The mission specification team applied best military judgement to specify required capability for collective tasks by specifying the minimum state of the platform and lower level collective tasks that comprise the collective task. Returning to the football analogy this approach is akin to the coach assessing the likelihood of success for the pass play discussed earlier based on the ability of the offensive line, receivers, and quarterback to perform their assigned tasks. He knows which of those lower level tasks can be Amber or Red (on a scale of Green = fully capable, Amber = partially capable, and Red = not capable) and which must absolutely be Green if the pass play is to have any chance of success.

To assess collective tasks for the demonstration, we:

- Analyzed each collective task to be simulated
- Decomposed each collective task into lower level collective and platform tasks
- Specified the minimum acceptable range of Green, Amber, Red states for that set of decomposed tasks
- Assigned a rating of Green, Amber, or Red to the collective task based on the actual state of the set of lower level collective and platform tasks

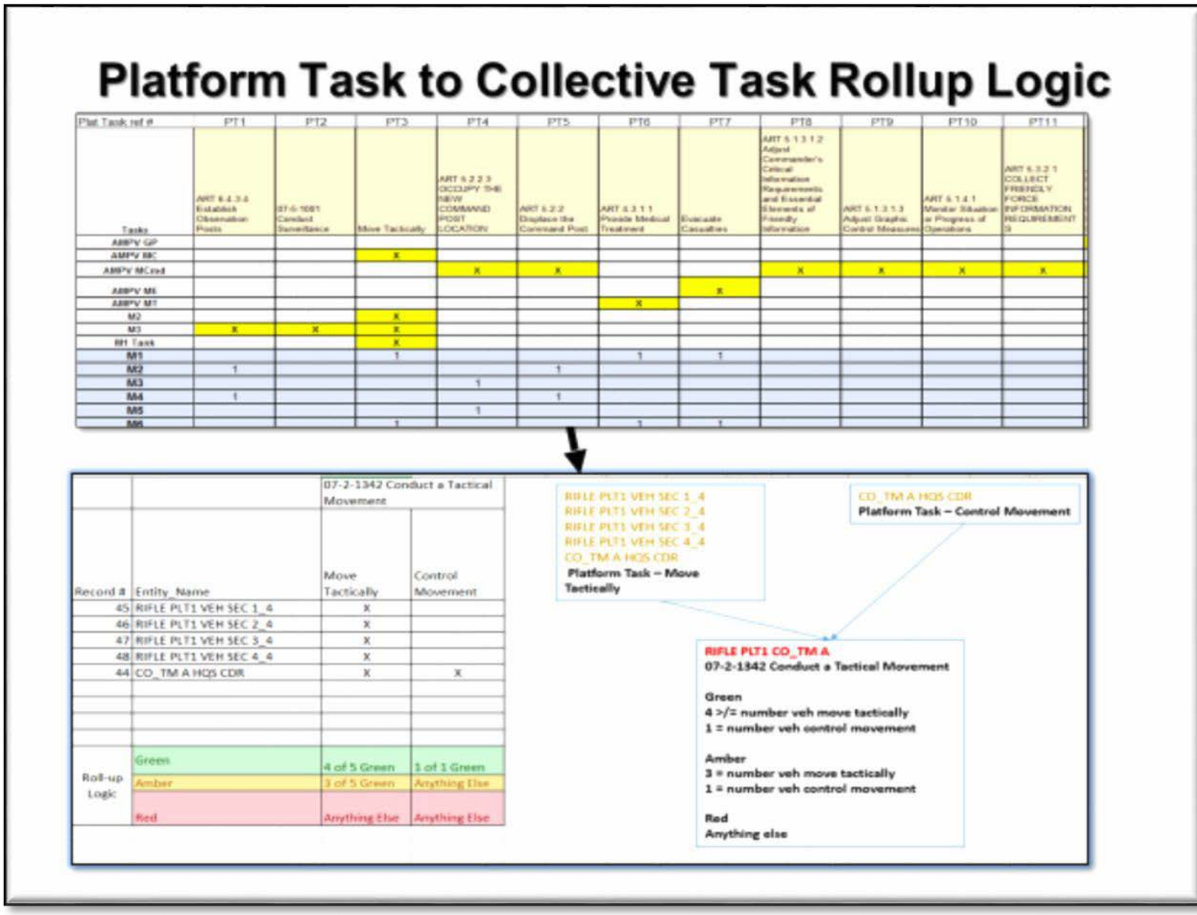


Figure 23 Platform Task to Collective Task Logic

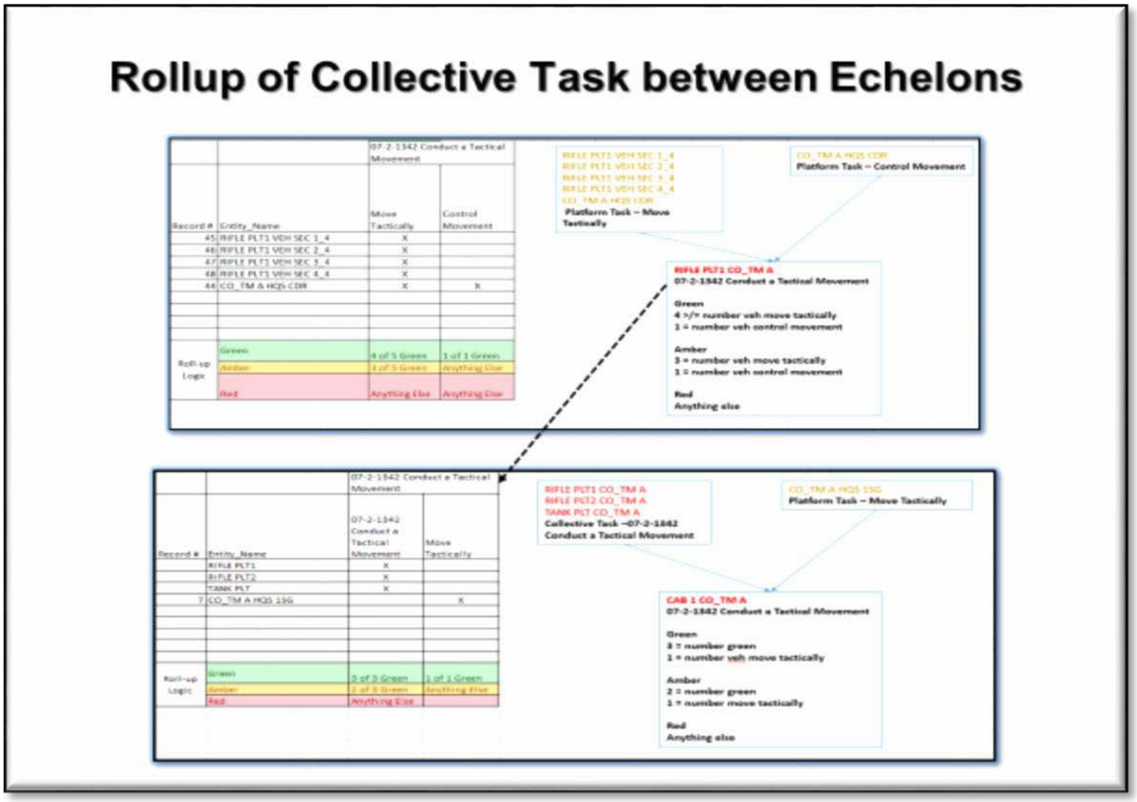


Figure 24 Lower Level Collective to Higher Echelon Collective Task

2.5.7 Specifying organizations and entities to be included in the model.

A single ABCT is structured vertically into four successively lower echelons (Battalion, CO/TM, Platoon, Squad/Section). Each echelon is structured horizontally with multiple tactical organizations (e.g., 3 Combined Arms Battalions, each with 4 Company/Teams and 1 Headquarters Company).

The team used Army Field Manual (FM) 3-96 (27) as the source for the ABCT organizational structure. In our demonstration instance of MUVES, only a relatively small set of organizations will be represented by platforms modeled. For that reason, we used a "soda straw" approach to determine which organizations from the MLS 2.0 scenario to include in our tactical vignettes and ultimately in our model "sand box".

The AMPV was selected as the primary system of interest for the demonstration. We ultimately decided to model AMPVs at every echelon from ABCT to squad/section because they are a relatively "low density" system in the Modified Table of Organization and Equipment (MTOE) for the ABCT. The other systems modeled (M1, M2, and M3) were included primarily to make the demonstration more interesting in terms of modeling the effect of kinetic interactions and setting the stage for vertical, multi-echelon assessment of resulting impact on mission effectiveness.

The team used the Heavy Brigade Combat Team (HBCT) organizational structure and platform authorizations contained in Fort Knox Supplemental Manual (FKSM) 71-8 (28) as the primary source to determine authorizations for the modeled systems. We substituted the appropriate AMPV variant for every authorization for a M113 Family of Vehicles system and selected other systems performing functions (e.g., Mission Command) for which the AMPV Mission Variants were designed. We maintained the FKSM 71-

8 specified authorizations for the other (Ground Combat Vehicle (GCV) – type) systems for those organizations included in our “soda straw”.

We applied the “soda straw” approach to demonstrate the ability to enable a “bottoms up” assessment of the operational impact of modeled kinetic and reliability failure interactions from component-to-platform-to-platform task and ultimately to collective task and mission up the brigade (ABCT) echelon. Figure 25 illustrates the selection of organizations from ABCT to CO/TM level that were modeled as either individual entities or collections of lower level entities.

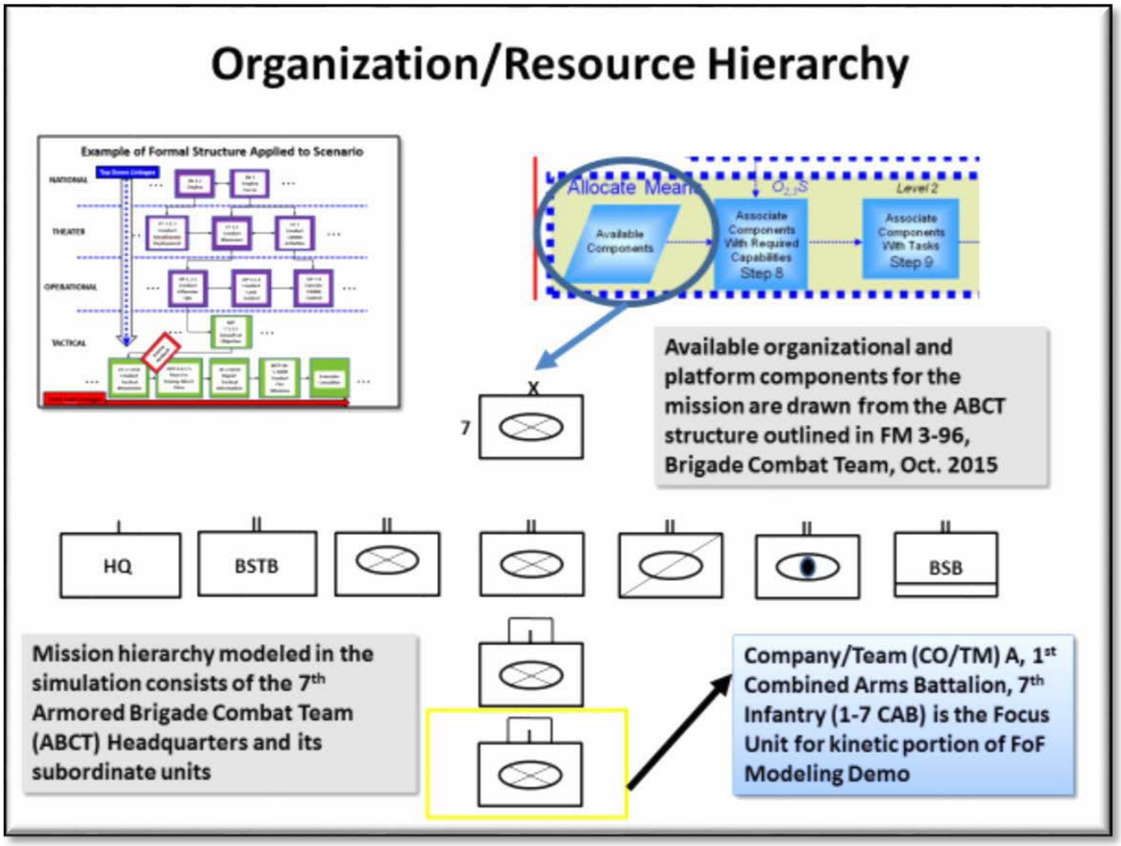


Figure 25 Organization Hierarchy Modeled in the Simulation

AMPV Mission Command variants were substituted for M113 FOV authorizations contained in the following organizations:

- ABCT Tactical Command Post
- CAB Command Post
- CAB Mortar Platoon Headquarters

AMPV Mortar Carrier variants were substituted for M113 FOV authorizations contained in the Mortar Platoon of the CAB’s Headquarters and Headquarters Company (HHC)

AMPV Medical Treatment and AMPV Medical Evacuation variants were substituted for M113 FOV authorizations contained in the medical platoon of the CAB’s HHC.

An AMPV General Purpose variant was substituted for the one M113 FOV authorization in the headquarters section of the mechanized Company/Team.

We included the authorizations for:

- 4 M1 Tanks from the Tank Platoon of the Mechanized Company/Team
- 9 M2 BFVs from the two Mechanized Platoons and the Command Section of the Mechanized Company/Team.
- 3 M3 CFVs from the Scout Platoon of the CAB

2.5.8 Development of the Time Ordered Event List (TOEL). Dynamically simulating the impact of kinetic interactions, reliability failures, and maintenance repair events required some means to generate events that cause these interactions over the course of mission execution. We elected to meet this requirement by developing a detailed TOEL to stimulate interactions. TOELs are nothing new to modeling and simulation, or for that matter to training, experimentation and operational testing applications. Part of our challenge, however, was developing a TOEL methodology that could capture the COA and the mission/exercise/test planner's intent with enough structure and detail to clearly communicate that intent to the model developer. Figure 26 illustrates the TOEL format used.

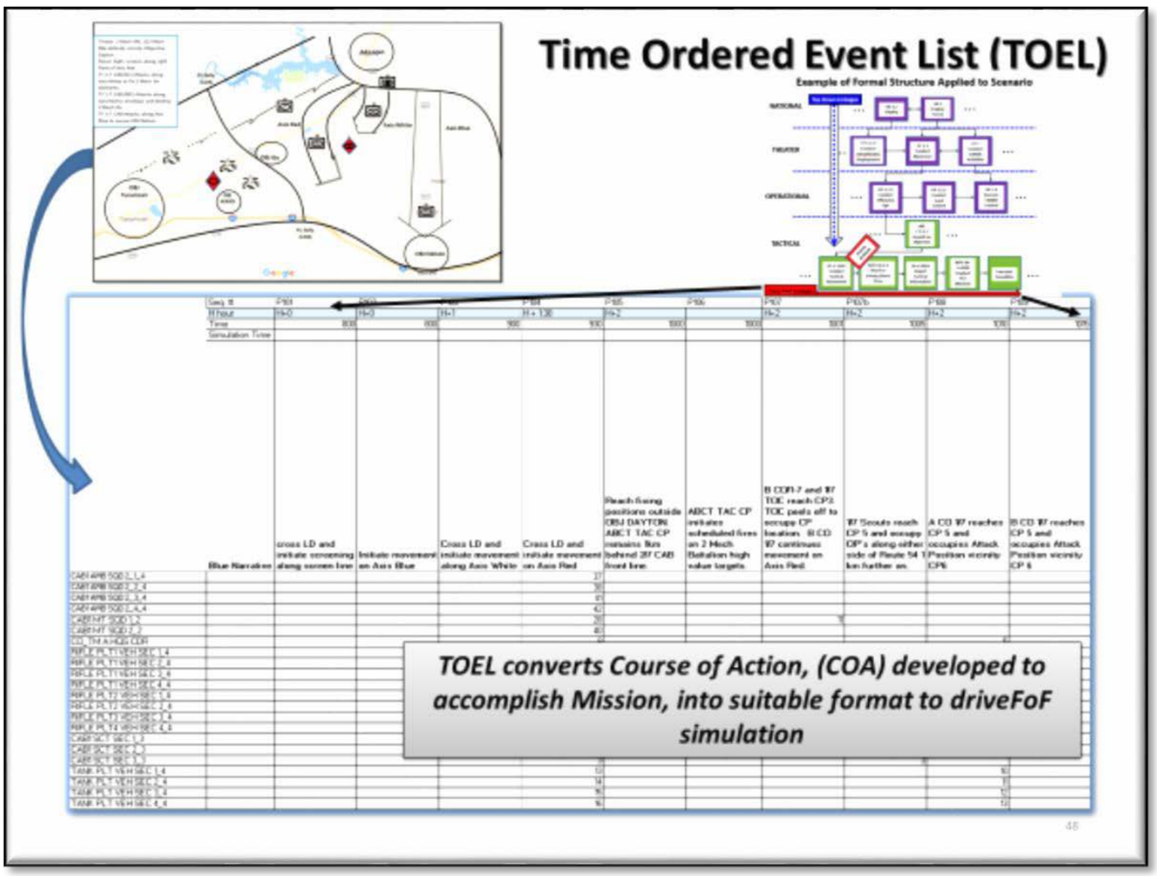


Figure 26 Time Ordered Event List

3 - SYSTEM SPECIFICATION AND DYNAMIC GEOMETRY

3.1 Major Steps for System Specification. The system specification team applied a four-step approach to developing the system specification for this demonstration:

1. Document system components, functions and capabilities for the AMPV-like candidate and a tank.

2. Develop an unclassified method to model component dysfunction and repair.

3. Generate MUVES model inputs for the two platform types (59 sets of model inputs) with required 'scene' and FoF 'sessions' files specific to the simulated scenario.

4. Code to map platform to capability to capability state using the method defined in step 2 above.

3.1.1 Document system components, functions and capabilities for the M113A3 and AMPV candidate concept. ARL SLAD engineers researched and documented information contained in readily available knowledge repositories within ARL and AMSAA.

3.1.2 Develop an unclassified method to model component dysfunction and repair. The SLAD engineer reviewed a failure modes and effects analysis (FMEA) of an unclassified AMPV-like platform and provided updates to an Excel workbook as the input file for the FMEA Input Review and Sysdef Tool (FIRST). FIRST requires an Excel based FMEA workbook to create the MUVES input files. The workbook contains 5 tabs; association table, FMEA table, multiple vulnerable (MV) dependencies, evaluation methods (EM) mapping and inputs table. The association table maps BRL-CAD regions to MUVES components. The FMEA table identifies critical components and their vulnerability mode (i.e., capability). The MV dependencies tab is used to define the relationship between multiple vulnerable components (i.e., the engineered design relationship of components to functions and capabilities). The EM tab is used to map critical components to one or more evaluation method where required parameters are designated in the EM inputs tab. FIRST uses the computer aided design (CAD) geometry of the platform and FMEA workbook to develop MUVES inputs for modeling. One of the critical inputs to MUVES is the system definition (sysdef) file which describes the platform failure analysis logic trees (FALTs). Figure 27 provides a brief description of FIRST.

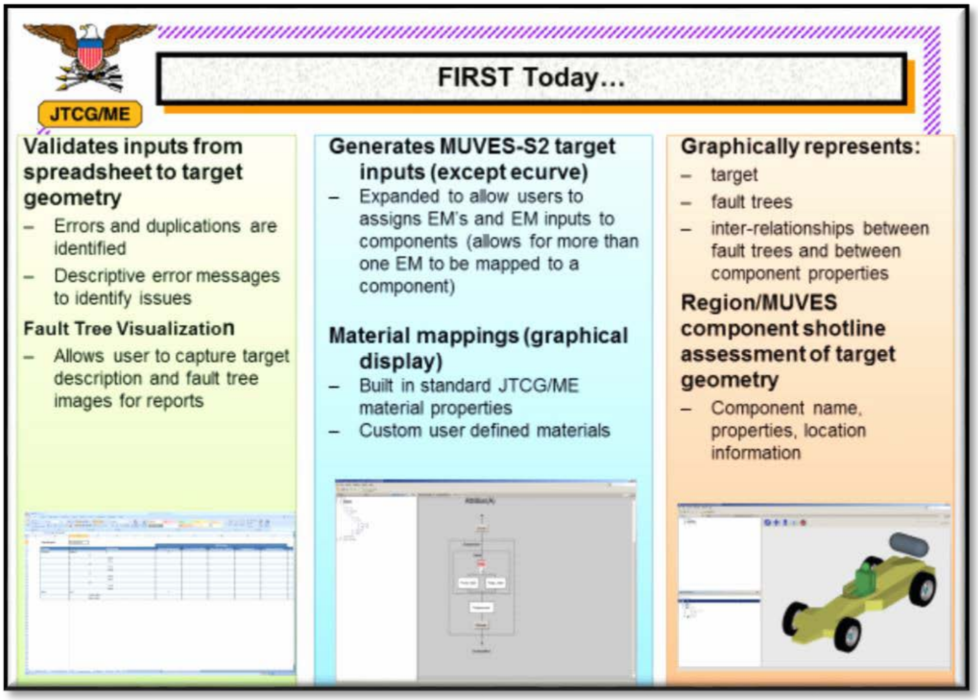


Figure 27 FMEA Input Review and Sysdef Tool (FIRST)

AMSAA provided the demonstration team with a proposed methodology for incorporating failure rates for critical components and mapping them to impacted system FALTs. Figure 28, below, illustrates the set of failure rates by major subsystem used for the AMPV system file.

Failure Rates by Major Subsystem for AMPV

Reliability Block	Total Sample Size	Total OP Miles	Total OP Hours	Total NMC Actions	NMC failures/mile	NMC failures/hour
ENGINE	383	48,503	14,557	254	0.0052	0.0174
OTHER (ELECTRICAL)	383	48,503	14,557	92	0.0019	0.0063
OTHER (FIRE FIGHTING)	383	48,503	14,557	5	0.0001	0.0003
OTHER (HULL/FRAME)	383	48,503	14,557	35	0.0007	0.0024
Other (APU)	383	48,503	14,557	71	0.0015	0.0049
OTHER (MISC)	383	48,503	14,557	55	0.0011	0.0038
Other (STEERING)	383	48,503	14,557	8	0.0002	0.0005
SUSPENSION	383	48,503	14,557	66	0.0014	0.0045
SUSPENSION (roadwheel only)	383	48,503	14,557	7	0.0001	0.0005
TRANSMISSION	383	48,503	14,557	73	0.0015	0.0050

Figure 28 Failure Rates by Major Subsystem for AMPV

AMSAA also provided the team with a description of the component repair logic used for Joint Light Tactical Vehicle (JLTV) availability modeling (figure 29). JLTV data was provided in lieu of AMPV data, which was not available at the time of the study. These critical AMSAA inputs enabled the demonstration team to incorporate and account for reliability failures and repair events in the dynamic geometry portion of the integrated modeling environment.

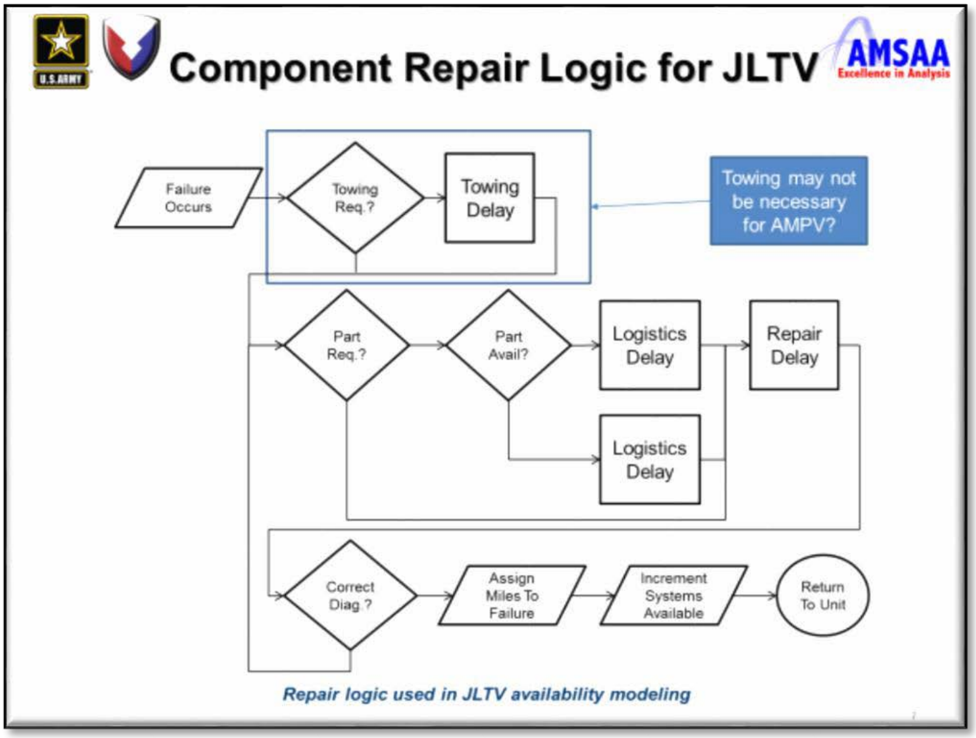


Figure 29 Repair Logic for JLTV

The proof of principle was focused on analyzing the AMPV and the effects of platform degradation on mission effectiveness. The M1 Tank, M2 BFV and M3 CFV were also played for the tank and mechanized infantry platoons of one CO/TM, and the Scout Platoon of one CAB. At the request of the mission specification team, existing products from a previous criticality analysis of the M1 tank were leveraged for the second platform modeled in MUVES. Both the M1 tank and AMPV were modeled in BRL-CADTM for use in MUVES. The multiple AMPV mission variants and platforms were modeled from one CAD geometry file (Figure 30).

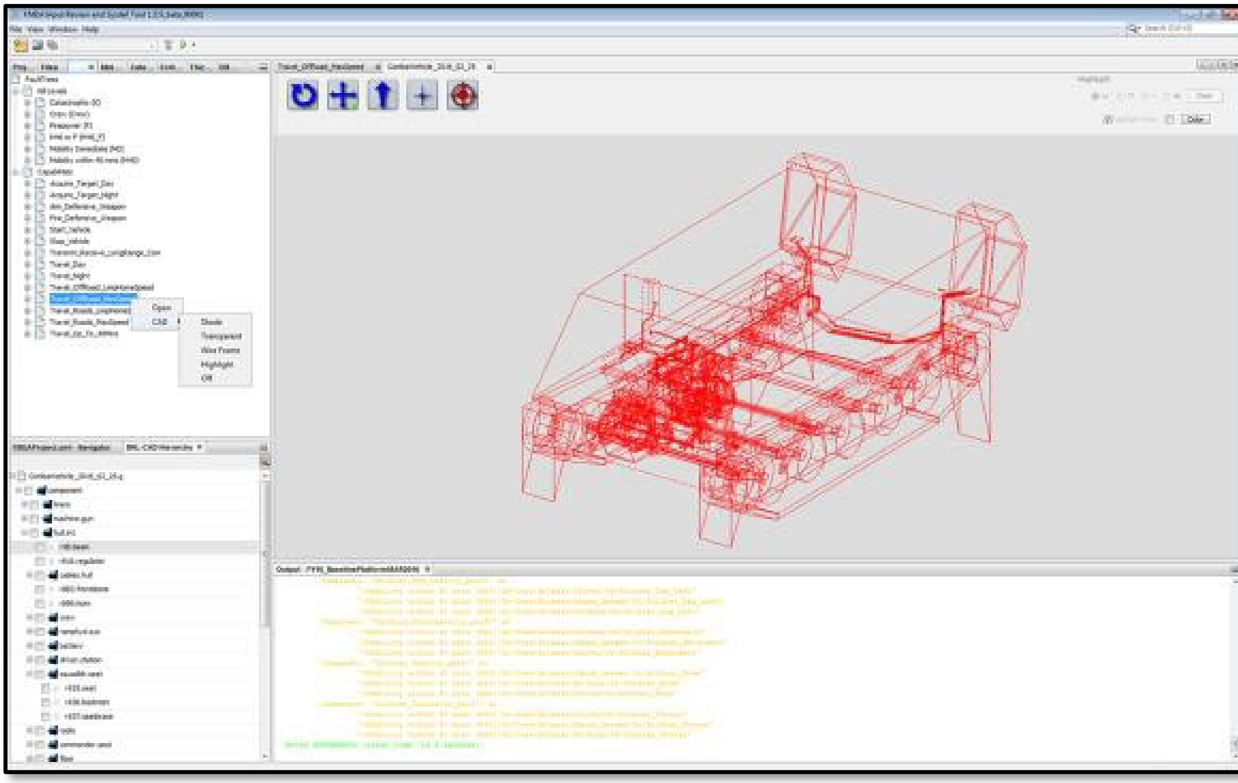


Figure 30 Unclassified AMPV Geometry File

3.1.3 Generate MUVES inputs for two platforms. The sysdef files created in FIRST provided the inputs for MUVES (Figure 31). The MUVES team used the sysdef file to define and visualize the system representation for modeling, assign evaluation methods to critical components, and generate MUVES model inputs. MUVES inputs included ten input files (ccmap, des, ecurve, matprop, prop, scene, states, sysdef, target, and threat) required by MUVES to model each platform. MUVES is a well understood and well documented system vulnerability and lethality analysis model used widely by ARL SLAD to support live fire test and evaluation. It is normally limited to modeling and analyzing results of kinetic interactions generated by impact from a wide range of munitions. We deliberately limited ballistic interactions used for the demonstration to 2 types of munitions representing direct fire and indirect fire engagements to remain at the unclassified level and to simplify the model for demonstration purposes.

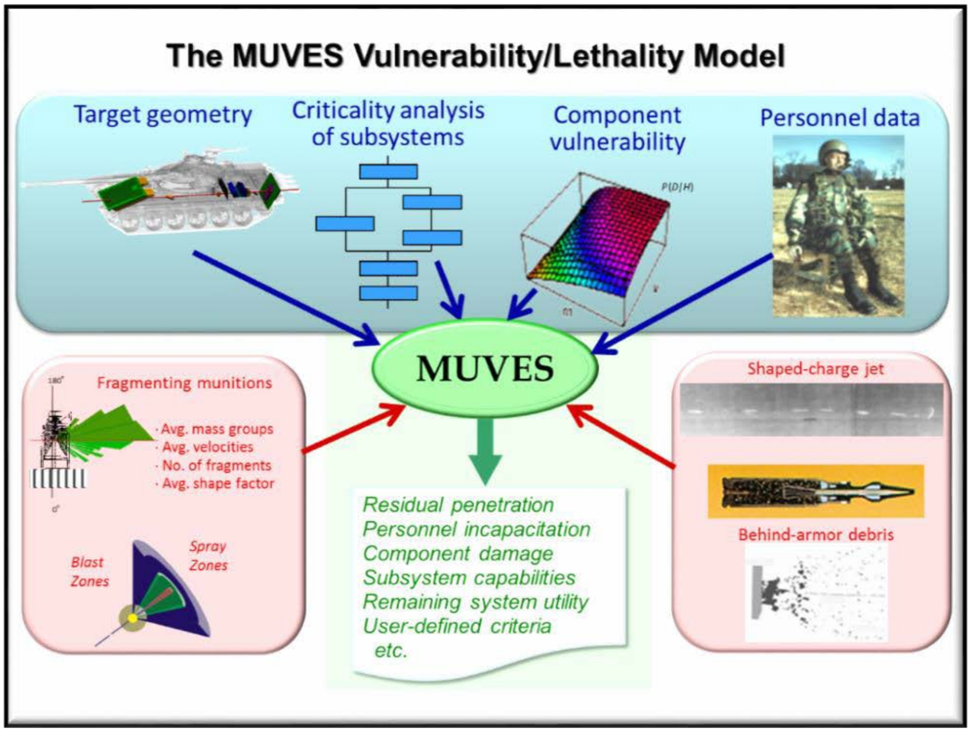


Figure 31 The MUVES Software Package

3.2 Application of MUVES to generate dynamic geometry.

3.2.1 Overview. For demonstration purposes MUVES provided the V/L service to generate and retain component damage, maintain platform status (MMF Level 2 data), and transmit resulting platform capability (MMF Level 3 data) in response to messages received from the MMF FoF model via an API. See Figure 35 for a conceptual overview of the MMF Dynamic Assessment Suite and the relationship between the MMF FoF model and MUVES. Figure 32 below illustrates the major components of MUVES as used for this demonstration. Note that the API, highlighted in bold type, had to be jointly developed by the MUVES and MMF FoF model teams and was the key enabler for dynamic exchange of data between the two models. The API will be discussed in greater detail in Appendix B of this report.

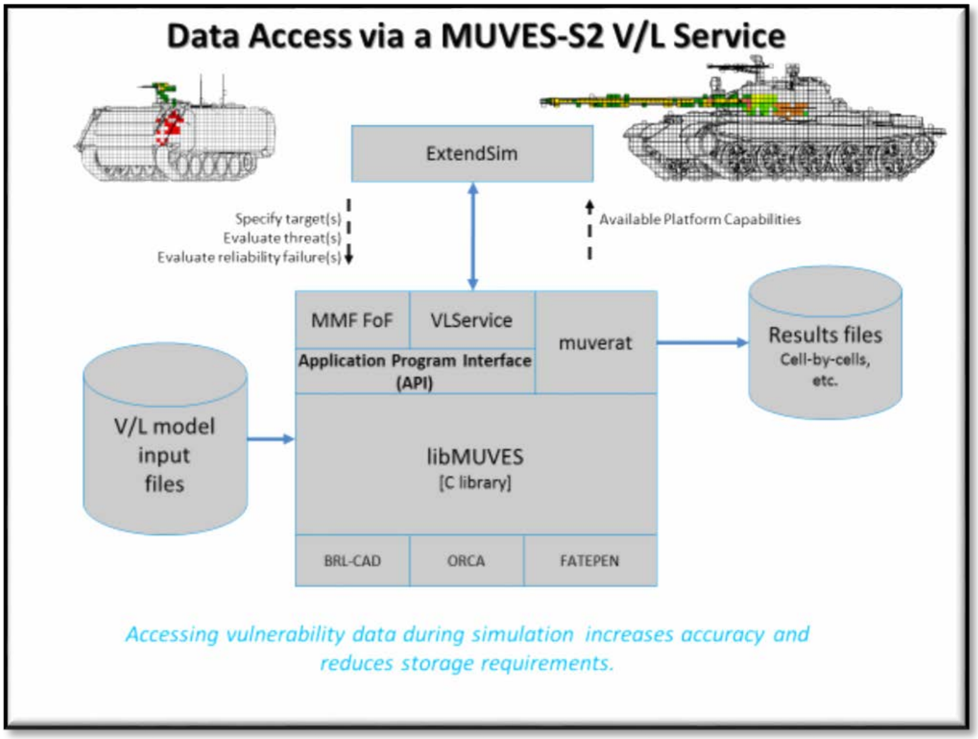


Figure 32 MUVES V/L Service

In execution the MUVES MMF model leveraged pre-established connections between the MMF Level 2 platform components included in the system model and the MMF Level 3 functions they enabled. Figure 33, below, illustrates this connection.

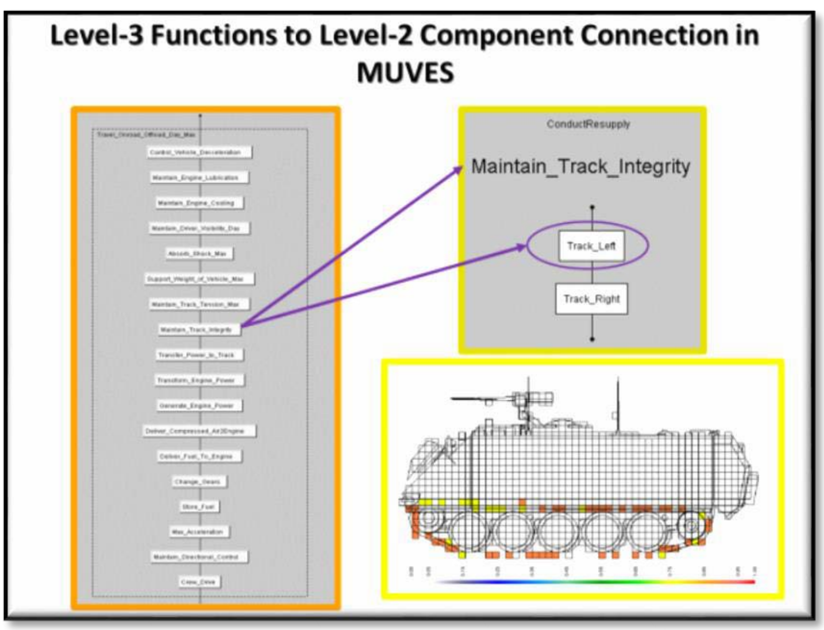


Figure 33 Level 3 Functions to Level 2 Components in MUVES

In the same way, the model applied the engineered design connections between component functions and platform capabilities and between Level 3 platform capabilities and Level 4 task requirements. See Figure 34.

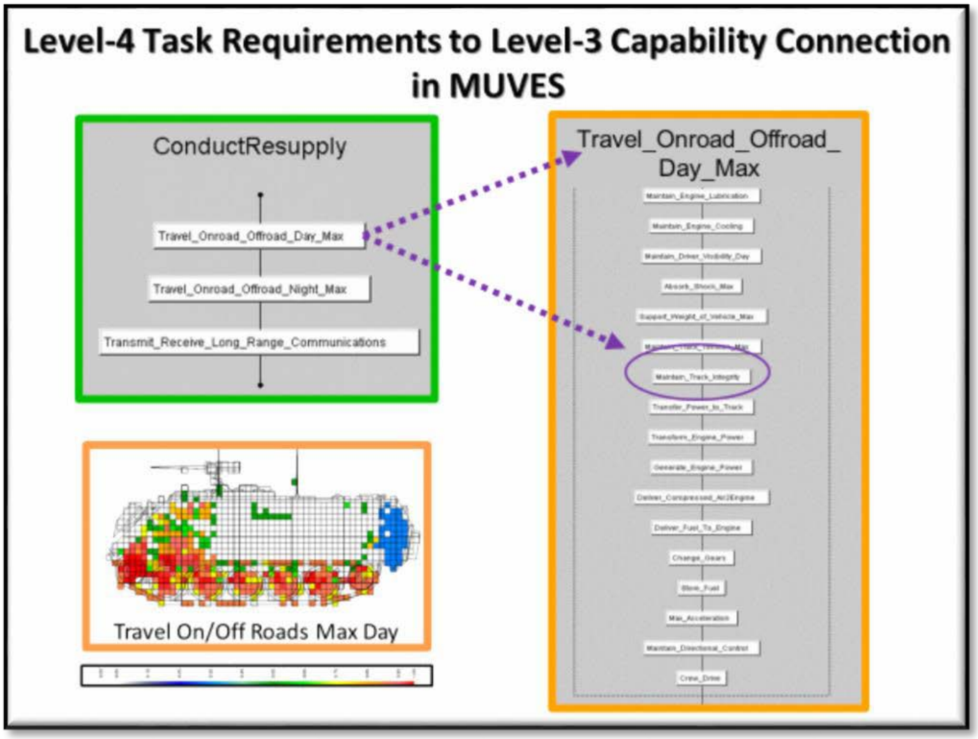


Figure 34 Level 4 Task Requirements to Level 3 Capabilities

4 - Model Overview

4.1 MMF Dynamic Assessment Suite. Figure 35 provides a high level conceptual view of the MMF Dynamic Assessment Suite model. We used ExtendSim© as the simulation development environment to develop the MMF FoF model at the core of the MMF Dynamic Assessment Suite. The MMF FoF model instantiated the mission specification products, (including formal task structures), discussed in Section 2, Mission Specification, and enabled simulation of mission execution and assessment. The MUVES MMF model provides dynamic geometry using detailed models for the platform entities of interest (AMPV, M1, M2, and M3) and determining effects of ballistic and material reliability failure interactions in terms of platform state changes. The MUVES MMF model further computes resulting capability states for affected platforms and returns the state of the platform Level-3 capabilities to the MMF FoF model via the MMF Dynamic Assessment Suite API. Development of the API required a joint contractor/government team effort to enable the exchange of data between the MMF FoF and MMF MUVES models. See Appendix B for a detailed description of the MMF FoF model and the API.

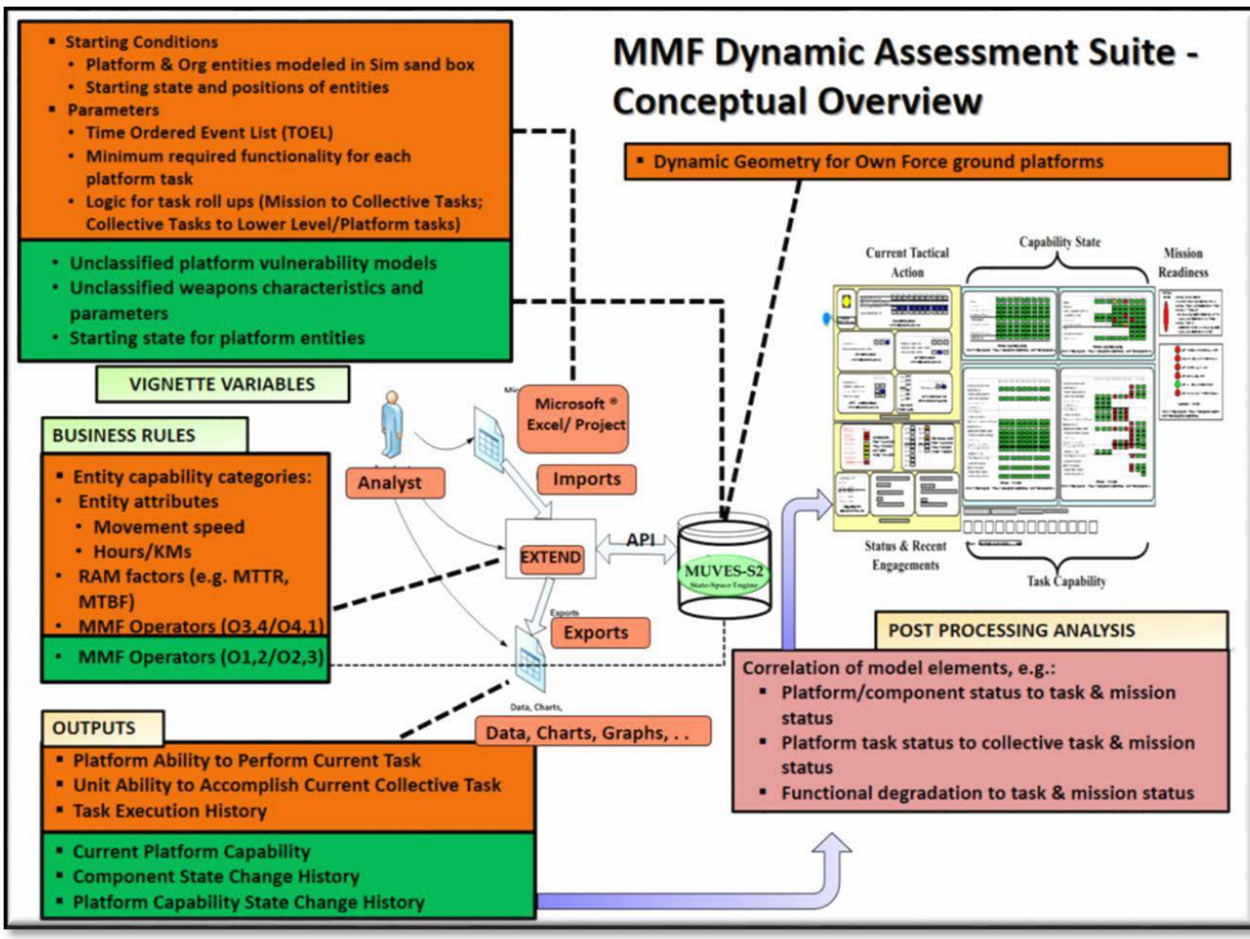


Figure 35 Conceptual View of the MMF Dynamic Assessment Suite model

5 - Model Execution and Results

5.1 Model Output Structure

From each replication of the model, we collected data that provided insight into the ability of the individual platforms and higher level units to perform relevant tasks at six selected points during the operation. These six points corresponded to significant events enumerated by the TOEL in the Deliberate Attack Exploitation phase, named for reference purposes: P104, P107, P108, P110, P114, and P115. For analysis, we selected platform and collective tasks that were relevant for execution during those events. Each event differed in terms of the time and location (MMF Level 5); tasks being executed (MMF Level 4); entities involved (MMF Level 2) and the interactions occurring (MMF Level 1). Accordingly, we focused on the specific platforms and units involved in the interactions at those event points and evaluated their ability to perform the subset of the 24 platform tasks and 18 collective tasks that were relevant to mission accomplishment at that event point or to immediate, follow-on operations. Run results for each event as the platform and collective task levels may be requested from the authors.

At each of the evaluated event points, the model captured relevant data in its database and, after the run, exported the data into MS Excel workbook files for later data analysis.

- TaskCapCol.xlsx – This file contained the collective task assessment for each of the 10 studied units at each of the event points. Table 1 shows an example extract of the collective task assessment table for one event point in one replication of the simulation model. Note that only a subset of units and tasks were evaluated, those units involved at that event point and those tasks relevant to the situation. An entry of "2" indicates that the unit was fully able to accomplish the task. An entry of "1" indicates the unit could only partially accomplish the task. An entry of "0" indicates that the unit was unable to accomplish the task.

Table 1 Example collective task assessment table - extract

Unit_Name	CTask1	CTask2	CTask3	CTask4	CTask5	CTask6	CTask7	CTask8	CTask9	CTask10	CTask11	CTask12
ABCT CAB1							1	0				
CAB1 TOC			1	0			1		2	0	2	
CAB1 MT PLT					2	1						
CAB1 MORT PLT												
CAB1 SCT PLT												

- TaskCapPlat.xlsx – This file contained the platform task assessment for each of the 59 individual platforms (entities) at each event point. Table 2 shows an example platform task assessment table for one event point in one replication of the simulation model. As with the collective task assessment, only the relevant subset of units and tasks were evaluated. An entry of "1" indicates that the platform was fully able to accomplish the task. An entry of "0" indicates that the platform was unable to accomplish the task.

Table 2 Example platform task assessment table - extract

Entity_Name	PTask1	PTask2	PTask3	PTask4	PTask5	PTask6	PTask7	PTask8	PTask9	PTask10	PTask11	PTask12	PTask13	PTask14	PTask15	PTask16
ABCT TAC CURRENT OPS 1 TACP																
ABCT TAC CURRENT OPS 2 INTEL																
CAB1 CP C4 OPS			0	0	0			1	1	1	1	1	1	1		0
CAB1 CP CURR OPS FS			0	0	0			1	1	1	1	1	1	1		0
CAB1 CP CURR OPS INTEL			0	0	0			0	0	0	0	0	0	0		0
CAB1 CP CURR OPS TACP			0	0	0			1	1	1	1	1	1	1		0
CAB1 CP CURR OPS1			0	0	0			1	1	1	1	1	1	1		0
CAB1 CP CURR OPS2			1	1	1			1	1	1	1	1	1	1		1
CAB1 CP SUST			0	1	0			0	0	0	0	0	0	0		0
CAB1 MTR PLT HQ																
CAB1 AMB SQD 1_1_4																
CAB1 AMB SQD 1_2_4																
CAB1 AMB SQD 1_3_4			1				1								0	1
CAB1 AMB SQD 1_4_4			1				1								0	1

In addition to the task assessments, the model produced a log file for each type interaction event - reliability events, ballistic interactions, or repair events – and exported the data into MS Excel workbook files for later data analysis. For reliability events, the logs were combined into one file, MaintFailEventData.xlsx, that contained the following logs:

- MaintFailEventDLLData – This log contained the name of the entity that was involved in each event, the type of system failure, and the capability assessment of the platform following the interaction against the 19 different capability levels. Table 3 contains an extract of the DLL Data for one replication of one event.

Table 3 Example MaintFailEventDLLData - extract

Entity_Name	Fail_Types	Lethality1	Lethality2	Communication1	Communication2	Protection1	Protection2	Protection3	Protection4	Protection5	Mobility1	Mobility2	Mobility3
ABCT TAC CURRENT OPS 1 FIRE SPT	MISC	1	1	1	1	1	1	1	1	1	0	1	1
ABCT TAC CURRENT OPS 2 INTEL	Engine	1	1	1	1	1	1	1	1	1	0	0	0
ABCT MAIN PROT	Elec	1	1	0	0	1	1	1	1	1	1	1	1
ABCT MAIN MVMT & MVR	Trans	1	1	1	1	1	1	1	1	1	0	0	0
ABCT MAIN FIRE SPT	Elec	1	1	0	0	1	1	1	1	1	1	1	1
ABCT MAIN PLANS	Engine	1	1	1	1	1	1	1	1	1	0	0	0
ABCT TAC CURRENT OPS 1 FIRE SPT	APU	1	1	1	1	1	1	1	1	1	0	0	0
ABCT TAC CURRENT OPS 1 TACP	Engine	1	1	1	1	1	1	1	1	1	0	0	0
ABCT TAC CURRENT OPS 2 INTEL	Trans	1	1	1	1	1	1	1	1	1	0	0	0
CAB1 SCT SEC 2_3	Engine	1	1	1	1	1	1	1	1	1	0	0	0
RIFLE PLT2 VEH SEC 2_4	APU	1	1	1	1	1	1	1	1	1	0	0	0
TANK PLT VEH SEC 1_4	APU	1	1	1	1	1	1	1	1	1	0	0	0
TANK PLT VEH SEC 2_4	Trans	1	1	1	1	1	1	1	1	1	0	0	0
TANK PLT VEH SEC 4_4	MISC	1	1	1	1	1	1	1	1	1	0	0	1
CO_TM A HQS 1SG	Elec	1	1	0	0	1	1	1	1	1	1	1	1
CAB1 CP C4 OPS	Engine	1	1	1	1	1	1	1	1	1	0	0	0
CAB1 CP CURR OPS INTEL	Elec	1	1	0	0	1	1	1	1	1	1	1	1
CAB1 CP CURR OPS1	APU	1	1	1	1	1	1	1	1	1	0	0	0
CAB1 MTR PLT HQ	MISC	1	1	1	1	1	1	1	1	1	0	1	0
CAB1 MTR SQD 3_4	Engine	1	1	1	1	1	1	1	1	1	0	0	0

- Log_MaintRecovery - In addition to the name of the entity involved in the interaction, this log identified the location of the event (EventPoint_Name), the type of platform involved, when platform recovery from the event started, and the amount of time taken for platform recovery. Note, only those reliability failures that resulted in a loss of mobility for the platform required recovery. Table 4 contains an extract of the Log_MaintRecovery table for one replication of the simulation at one event point.

Table 4 Example Log_MaintRecovery table - extract

RecoveryStart_Time	EventPoint_Name	Entity_Name	Entity_Type	Recovery_Time
257.1135399	E	CAB1 MTR PLT HQ	AMPV_MCmd	26.06950278
304.9159664	E	ABCT TAC CURRENT OPS 1 FIRE SPT	AMPV_MCmd	0
321.7436975	C	ABCT MAIN PROT	AMPV_MCmd	0
304.9159664	E	ABCT TAC CURRENT OPS 2 INTEL	AMPV_MCmd	30.02292505
321.7436975	C	ABCT MAIN MVMT & MVR	AMPV_MCmd	29.12999719
413.2310924	E	ABCT MAIN FIRE SPT	AMPV_MCmd	0
413.2310924	E	ABCT MAIN PLANS	AMPV_MCmd	28.7209305
1542.06378	W	ABCT TAC CURRENT OPS 1 TACP	AMPV_MCmd	30.09323358
1542.01378	W	ABCT TAC CURRENT OPS 1 FIRE SPT	AMPV_MCmd	32.25837632
1574.955512	I	TANK PLT VEH SEC 4_4	M1 TANK	0
1542.11378	W	ABCT TAC CURRENT OPS 2 INTEL	AMPV_MCmd	33.64281433
1575.955512	I	CO_TM A HQS 1SG	AMPV_GP	0
1583.034252	H	CAB1 CP CURR OPS INTEL	AMPV_MCmd	0
1574.955512	I	TANK PLT VEH SEC 2_4	M1 TANK	21.95156114
1571.955512	I	CAB1 SCT SEC 2_3	CFV	28.84832506
1574.955512	I	TANK PLT VEH SEC 1_4	M1 TANK	28.56784086

For ballistic interactions, the applicable logs were similarly combined into one file, BallisticEventData.xlsx, which contain the following:

- BallisticEventDLLData – This table contained the ballistic event parameter data for each of the ballistic interactions as well as the capability assessment of the platform following the interaction. The ballistic event parameters included an identification of the threat munition type involved, the weapons system, the angle the munition impacts the platform, and the impact point (point of aim) on the vehicle. Table 5 contains an extract of the BallisticEventDLLData table for one replication of the simulation at one event point.

Table 5 Example BallisticEventDLLData table - extract

Entity_Name	MunitionType	Range	AngleX	AngleY	PointOfAimX	PointOfAimY	PointOfAimZ	Lethality1	Lethality2	Communic	Communic	Protection	Protection	Protection	Prote	ction Protection	Mobility1	Mobility
CAB1 MT SQD 1_2	1	134.0196	158.8715	91.36829	0	1109.11197	862.112072	1	1	1	1	1	1	1	1	1	1	1
RIFLE PLT2 VEH SEC 1_4	1	125.5688	157.8655	88.8791	0	1425.75184	1296.45458	1	1	1	1	1	1	1	1	1	1	1
CAB1 CP SUST	1	134.0036	161.7039	90.49342	0	1632.42444	1132.19098	1	1	0	0	1	1	1	1	1	1	1
CAB1 MTR SQD 2_4	1	129.6715	158.8241	93.80143	0	1467.7242	1545.19918	1	1	1	1	1	1	1	1	1	1	1
TANK PLT VEH SEC 1_4	1	135.605	159.2127	91.27764	0	1213.16767	1148.95714	1	1	1	1	0	1	1	1	1	0	0
CAB1 CP CURR OPS TACP	1	131.5438	161.7957	92.97798	0	1494.06854	977.419461	1	1	1	1	1	1	1	1	1	0	0
CAB1 AMB SQD 2_2_4	1	124.1564	162.4754	87.41938	0	1526.98374	1108.99303	1	1	1	1	1	1	1	1	1	0	0
ABCT MAIN SUST	1	87.13121	86.50712	92.06746	3091.1007	0	1127.85329	1	1	1	1	1	1	1	1	1	1	1
CAB1 MT SQD 2_2	1	128.7002	155.7867	91.0393	0	860.404675	898.04684	1	1	1	1	1	1	1	1	1	0	0
ABCT TAC CURRENT OPS	1	130.2504	163.0521	88.42186	0	1491.94501	1300.86435	1	1	1	1	1	1	1	1	1	1	1
ABCT MAIN CURRENT OPS	1	82.43534	70.52865	92.5034	0	2463.46692	892.338724	1	1	1	1	1	1	1	1	1	1	1
ABCT MAIN PROT	1	81.42138	70.40846	88.93183	0	1108.89695	785.738269	1	1	0	0	1	1	1	1	1	1	1
ABCT MAIN PLANS	1	130.3709	159.8025	88.91659	0	1722.82673	1044.7203	1	1	0	0	1	1	1	1	1	1	1
ABCT TAC CURRENT OPS 2 I	2	898.8232	273.9241	91.38026	3140.59925	1298.53156	1594.12195	1	1	1	1	1	1	1	1	1	0	0
CAB1 CP CURR OPS1	3	12508.66	352.9355	44.0203	2040.64422	1164.8634	1800	0	0	1	1	1	1	1	1	0	0	0
CAB1 AMB SQD 2_1_4	1	100.5373	250.8131	92.3564	2794.81438	3100	661.626238	1	1	1	1	1	1	1	1	1	0	0
CAB1 CP CURR OPS FS	3	12491.69	350.6779	43.57077	3232.75771	1985.60758	1800	0	0	1	1	1	1	1	1	0	0	0
CAB1 SCT SEC 2_3	2	1304.323	287.6433	89.91495	1411.72025	3100	1832.36506	1	1	1	1	1	1	1	1	1	1	1
RIFLE PLT1 VEH SEC 2_4	2	1303.011	290.1765	89.75293	2137.20112	3100	1462.31811	1	1	1	1	1	1	1	1	1	1	1

- Log_ – This log identified the location of the event (EventPoint_Name), the name and type of platform involved, when the event occurred, and the amount of time taken for platform recovery if required. Table 6 contains an extract of the Log_ table for one replication of the simulation at one event point.

Table 6 Example Log_BallisticEvent

BallisticEventStart_Time	EventPoint_Name	Entity_Name	Entity_Type	Recovery_Time
256.2563025	F	CAB1 MTR SQD 2_4	AMPV_MC	0
295.1722689	B	ABCT MAIN SUST	AMPV_MCmd	0
285.0482197	F	CAB1 CP CURR OPS TACP	AMPV_MCmd	25.94670019
286.0752296	F	CAB1 AMB SQD 2_2_4	AMPV_ME	26.85670071
280.6842221	F	TANK PLT VEH SEC 1_4	M1 TANK	36.08902898
334.9411765	F	ABCT TAC CURRENT OPS	AMPV_MCmd	0
309.3115617	F	CAB1 MT SQD 2_2	AMPV_MT	28.39652806
348.3151261	D	ABCT MAIN CURRENT OPS	AMPV_MCmd	0
348.3151261	D	ABCT MAIN PROT	AMPV_MCmd	0
443.2563025	F	ABCT MAIN PLANS	AMPV_MCmd	0
1574.727752	Y	ABCT TAC CURRENT OPS 2 INTE	AMPV_MCmd	36.6909766
1587.955512	J	CAB1 CP CURR OPS1	AMPV_MCmd	30.1615805
1634.029134	O	CAB1 SCT SEC 2_3	CFV	0
1634.129134	O	RIFLE PLT1 VEH SEC 2_4	BFV	0
1634.179134	O	RIFLE PLT1 VEH SEC 3_4	BFV	0

Likewise, applicable logs for repair events, were combined into one file, RepairEventData.xlsx.

- RepairEventDLLData – This table contained the entity name and the platform capability assessment pursuant to the event. Since repair events were designed to "completely repair" all shortcomings, the resulting logs displayed "1" in every capability category. The team used this log to verify that the repair event construct was working in the model as expected. Table 7 contains an extract of the RepairEventDLLData table for one replication of the simulation.

Table 7 Example RepairEventDLLData table - extract

Entity_Name	Lethality1	Lethality2	Communication1	Communication2	Protection1	Protection2	Protection3	Protection4	Protection5	Mobility1	Mobility2
TANK PLT VEH SEC 2_4	1	1	1	1	1	1	1	1	1	1	1
CAB1 MT SQD 2_2	1	1	1	1	1	1	1	1	1	1	1
ABCT BSTB MP TM 5_6	1	1	1	1	1	1	1	1	1	1	1
TANK PLT VEH SEC 4_4	1	1	1	1	1	1	1	1	1	1	1
ABCT BSTB CURRENT OPS OPS	1	1	1	1	1	1	1	1	1	1	1
RIFLE PLT1 VEH SEC 3_4	1	1	1	1	1	1	1	1	1	1	1
CAB1 SCT SEC 3_3	1	1	1	1	1	1	1	1	1	1	1
CAB1 AMB SQD 1_1_4	1	1	1	1	1	1	1	1	1	1	1
CAB1 AMB SQD 1_4_4	1	1	1	1	1	1	1	1	1	1	1
ABCT TAC CURRENT OPS 2 INTEL	1	1	1	1	1	1	1	1	1	1	1
ABCT MAIN INTEL	1	1	1	1	1	1	1	1	1	1	1
CAB1 AMB SQD 2_3_4	1	1	1	1	1	1	1	1	1	1	1

- Log_RepairEvent – This log is like the other event logs in that it identified the name and type of platform involved, when the event occurred, and the amount of time taken for platform repairs. Table 8 contains an extract of the Log_RepairEvent table for one replication of the simulation.

Table 8 Example Log_RepairEvent

RepairEventStart_Time	EventPoint_Name	Entity_Name	Entity_Type	Repair_Time
300.5630252	G	TANK PLT VEH SEC 2_4	M1 TANK	32.17652805
333.0543006	G	RIFLE PLT1 VEH SEC 3_4	BFV	29.91290282
306.5630252	G	CAB1 MT SQD 2_2	AMPV_MT	64.88837528
353.419704	G	CAB1 SCT SEC 3_3	CFV	24.78203874
332.8877615	G	ABCT BSTB CURRENT OPS OPS	AMPV_MCmd	56.83620887
327.1472932	G	TANK PLT VEH SEC 4_4	M1 TANK	67.31176203
326.131867	G	ABCT BSTB MP TM 5_6	AMPV_GP	79.60073116
386.359085	G	CAB1 AMB SQD 1_4_4	AMPV_ME	34.41543172
1724.391597	V	CAB1 CP CURR OPS TACP	AMPV_MCmd	43.21893571
1787.397224	V	RIFLE PLT4 VEH SEC 4_4	BFV	29.10172189

5.2 Model Output

The model output mirrored the analytical approach. Assessments were performed by both MUVES and ExtendSim cascading upwards from the internal sub-systems of individual platforms to the collective task assessment at the battalion level. This "rolling-up" of assessments provided a direct linkage between the component state of a sub-system (MMF Level 2) and the resulting impact on the ability of the organization to successfully perform tasks, level 4 (Figure 36).

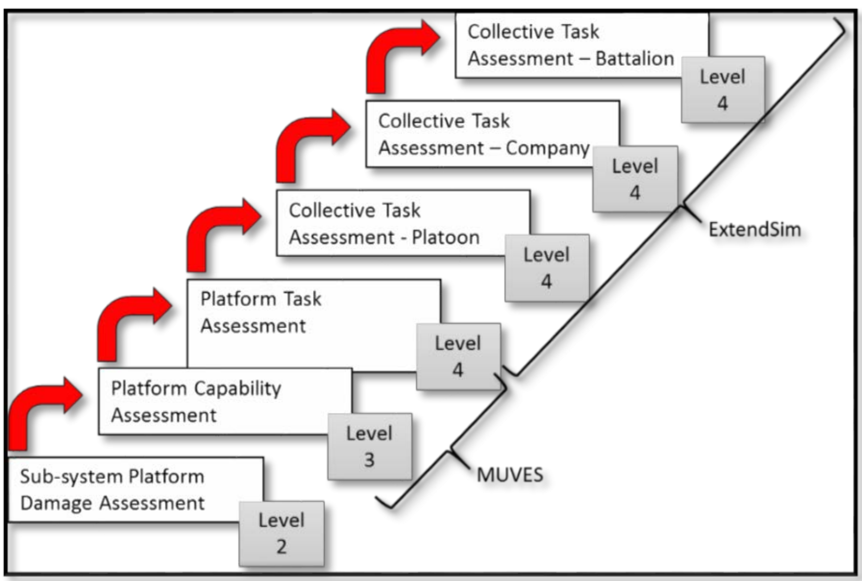


Figure 36 Assessment roll-up logic

For illustration, we will trace the logic in reverse for one task, beginning with a company-level, collective task assessment and work down. TOEL event P108 of the Deliberate Attack Exploitation phase described A/1-7 CAB reaching checkpoint 5 on Axis Red and occupying an attack positon vicinity checkpoint 6 Axis Red. Possible interactions leading up to this event involved potential reliability failures of the platforms prior to reaching checkpoint 3 (Figure 37).

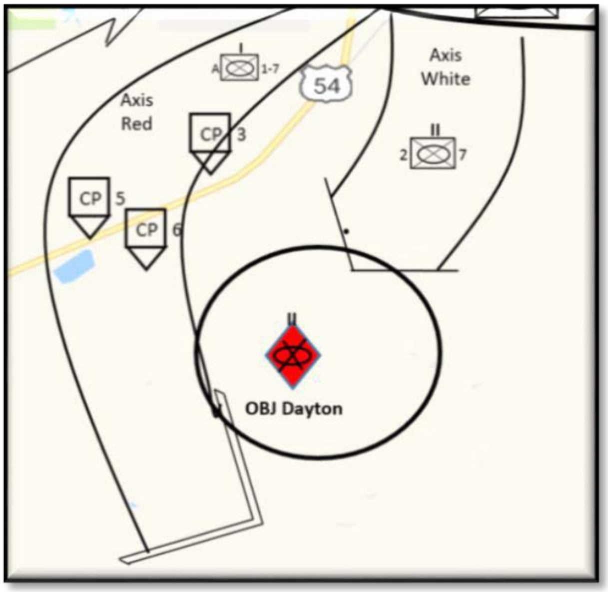


Figure 37 Deliberate Attack Exploitation Operational graphic - extract

As stated previously, not all collective tasks were relevant during all phases of the operation. At event point P108, the relevant task at the company level was collective task 12 - ART 1.5.2 Occupy an Attack and Assault Position. The company was supported by its platoons performing the same collective task at platoon level. For the company to be assessed fully capable of performing the task, all three of its subordinate platoons had to be fully capable (green) of performing the task. If two of the three platoons were assessed fully capable (green) of performing the task, the company would be assessed as amber, meaning the company was only partially capable of completing the task (Figure 38).

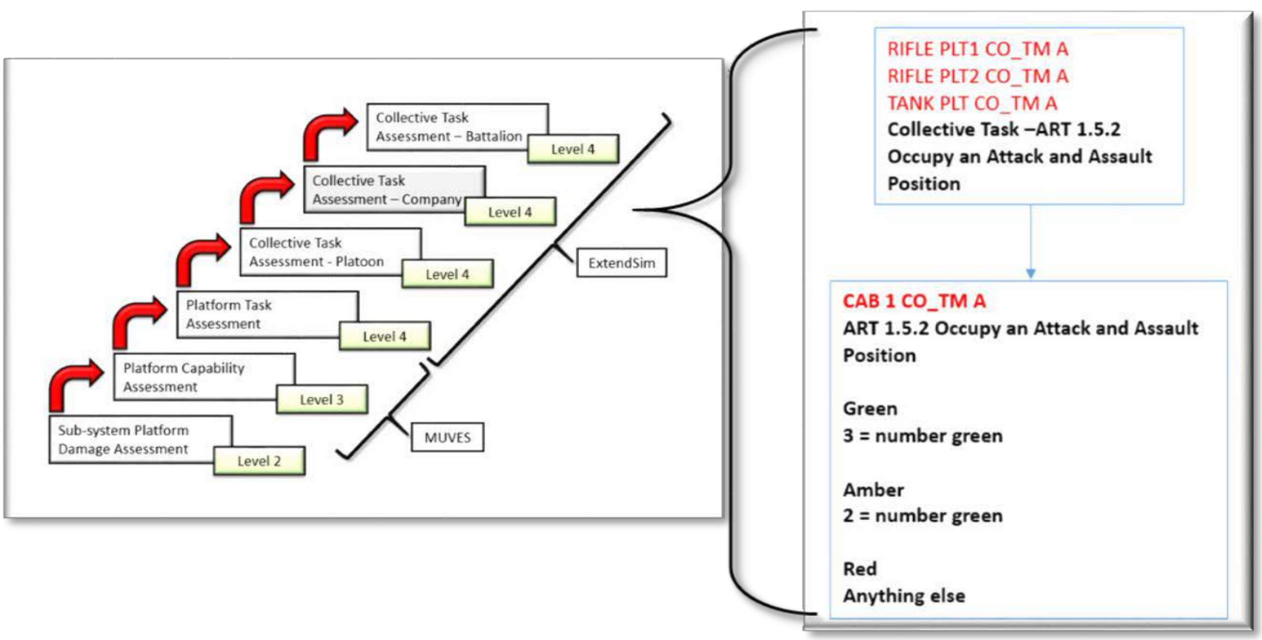


Figure 38 Assessment Logic - Company Collective Task 12

In this case, over the course of 30 replications of the simulation, the company was never fully capable of performing the task at event point P108. The company was assessed red across 90% of the simulation runs and amber across the remaining 10% of the simulation runs. Examining the collective task assessments for each of the platoons begins to reveal why. Rifle Platoon 1 was assessed green in only 23% of the 30 simulation runs. Rifle Platoon 2 was assessed green across only 37% of the simulation runs and the Tank Platoon was assessed green across only 10% of the simulation runs. (Figure 39).

CTask12			
Unit_Name	Green	Amber	Red
CAB1 CO_TM A	0%	10%	90%
CO_TM A RIFLE PLT 1	23%	63%	13%
CO_TM A RIFLE PLT 2	37%	13%	50%
CO_TM A TANK PLT	10%	20%	70%

Figure 39 Event Point P108 - Assessment Summary

There was not a single run when all three platoons were fully capable of performing the task. To illustrate the task roll-up logic, consider the collective task assessment for replication 20 of the simulation as shown in Figure 40 below. Because only one of the platoons was assessed as green on the task, the company was assessed as red, incapable of performing the task

Unit_Name	CTask12	Assessment
CAB1 CO_TM A	0	Red
CO_TM A RIFLE PLT 1	1	Amber
CO_TM A RIFLE PLT 2	0	Red
CO_TM A TANK PLT	2	Green

Figure 40 Replication 20, Event Point P108 Assessment

The platoon task assessment for collective task 12 depended on assessment of platform task 15, Occupy a Position for each modeled platform assigned to the platoon. Both rifle platoons, for example were composed of four individual sections in Bradley Fighting Vehicles (BFVs), for example RIFLE PLT1 VEH SEC 1_4, and any other modeled platforms temporarily attached to the platoon. Rifle Platoon 1 for example, included the company commander in his vehicle, CO_TM A HQS CDR, and Rifle Platoon 2 included the ambulance squad, CAB1 AMB SQD 1_1_4. The tank platoon's collective task assessment was similarly dependent on the individual platform assessment of its subordinate task sections in M1 tanks and an ambulance section (Figure 41).

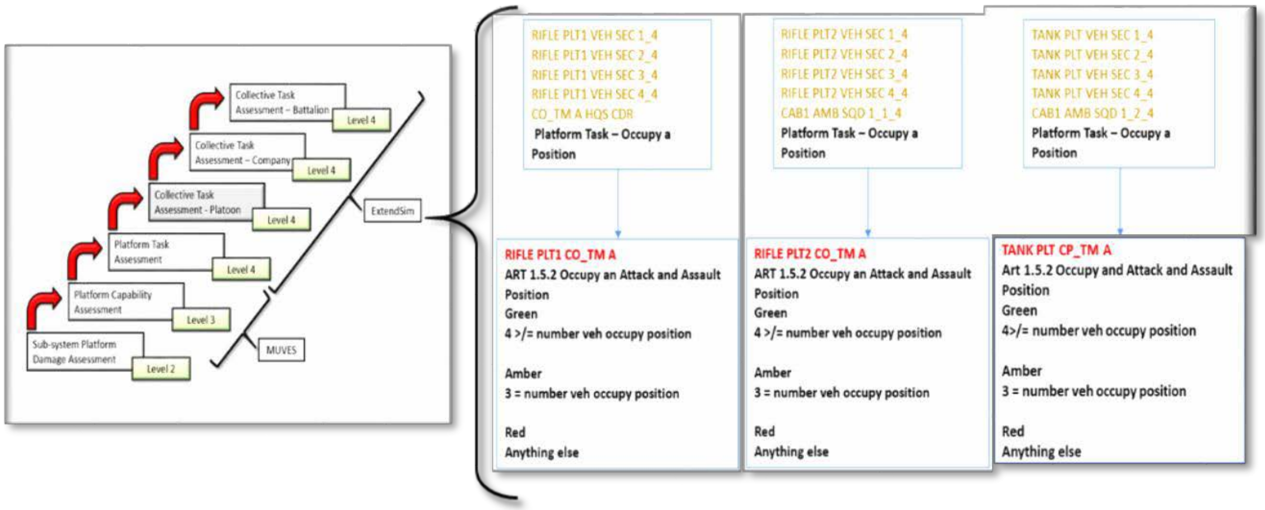


Figure 41 Assessment Logic - Platoon Collective Task 12

Figure 42 below shows the percentage of time each platform in the three platoons was assessed green across all 30 simulation runs for event point P108. Note that every platform across all three platoons was assessed green over less than 70% of the simulation runs.

RIFLE PLT1 CO_TM A			RIFLE PLT2 CO_TM A			TANK PLT CO_TM A	
RIFLE PLT1 VEH SEC 1_4	67%		RIFLE PLT2 VEH SEC 1_4	57%		TANK PLT VEH SEC 1_4	43%
RIFLE PLT1 VEH SEC 2_4	67%		RIFLE PLT2 VEH SEC 2_4	50%		TANK PLT VEH SEC 2_4	40%
RIFLE PLT1 VEH SEC 3_4	63%		RIFLE PLT3 VEH SEC 3_4	53%		TANK PLT VEH SEC 3_4	37%
RIFLE PLT1 VEH SEC 4_4	57%		RIFLE PLT4 VEH SEC 4_4	53%		TANK PLT VEH SEC 4_4	33%
CO_TM A HQS CDR	57%		CAB1 AMB SQD 1_1_4	63%		CAB1 AMB SQD 1_2_4	50%

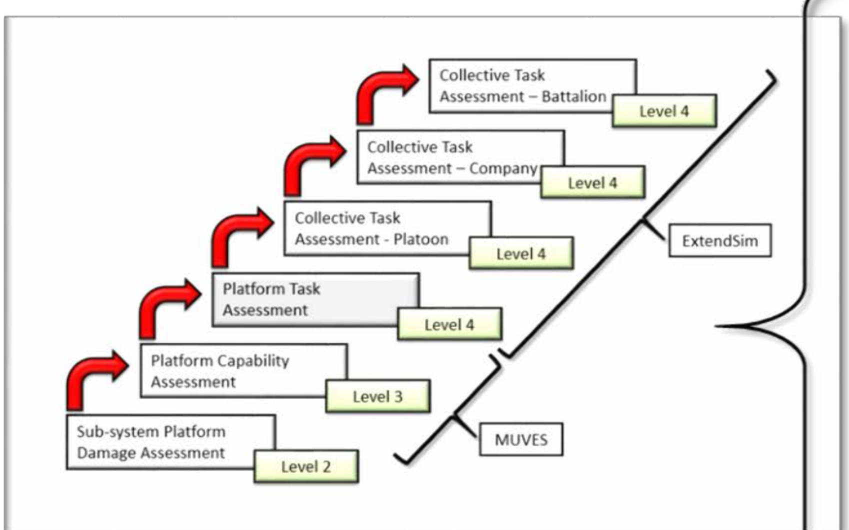
Figure 42 Event Point P108 - Assessment Summary

Figure 43 below shows the Platform task assessment for RIFLE PLT1 for replication 20 of 30 total replications. Because 3 of the 5 platforms were assessed green on the required platform task, platoon collective task 12 was assessed as amber.

Unit	CTask12	Asssessment
RIFLE PLT1 CO_TM A	1	Amber
Platform	PTask15	Assessment
RIFLE PLT1 VEH SEC 1_4	1	Green
RIFLE PLT1 VEH SEC 2_4	1	Green
RIFLE PLT1 VEH SEC 3_4	0	Red
RIFLE PLT1 VEH SEC 4_4	1	Green
CO_TM A HQS CDR	0	Red

Figure 43 Replication 20, Event Point P108 Assessment

The assessment for an individual platform task was completed by comparing the required capability states for each platform task, (determined during mission specification), to the residual capability state of the platform at the time of task execution. This is the critical point in the analysis where the connection between level 4 and level 3 of the MMF was made. As stated previously, the capability states of the platforms were described in terms of 19 individual capabilities: Lethality 1 and 2; Communication 1 and 2; Protection 1 through 5; and Mobility 1 through 10. For platform task 15, the required capabilities were: Mobility 3, Mobility, 5, Mobility 8, Mobility 10, and Communication 2; each of those capabilities must be present for the platform to be green for the task (Figure 44).



Plat Task ref #	PT15
Tasks	Occupy a Position
M1	
M2	
M3	1
M4	
M5	1
M6	
M7	
M8	1
M9	
M10	1
L1	
L2	
C1	
C2	1
P1	
P2	
P3	
P4	
P5	

Figure 44 Assessment Logic - Platform Task 15

The platform task assessment was dependent on the capability state of the platform at that event point, in this case Event Point P108. From the TOEL, the last interaction prior to the event point that would result in a state change to the platform was a reliability event. The model logic assessed the platform task by determining how many of the required capabilities for the task were provided by each platform entity.

Figure 45 below shows the capability assessment, with resultant task assessment for platform task 15, during replication 20 immediately prior to P108 for the platforms comprising RIFLE PLT1.

Entity_Name	Communication2	Mobility3	Mobility5	Mobility8	Mobility10	PTask15	Assessment
RIFLE PLT1 VEH SEC 1_4	1	1	1	1	1	1	Green
RIFLE PLT1 VEH SEC 2_4	1	1	1	1	1	1	Green
RIFLE PLT1 VEH SEC 3_4	1	0	1	0	1	0	Red
RIFLE PLT1 VEH SEC 4_4	1	1	1	1	1	1	Green
CO_TM A HQS CDR	1	0	0	0	0	0	Red

Figure 45 Replication 20, Event Point P108 Assessment - Capability to Platform Task 15

The capability of the individual platforms was dependent upon the status of the individual platform subsystems and how they are able or not able to interoperate to deliver the capability (Figure 46).

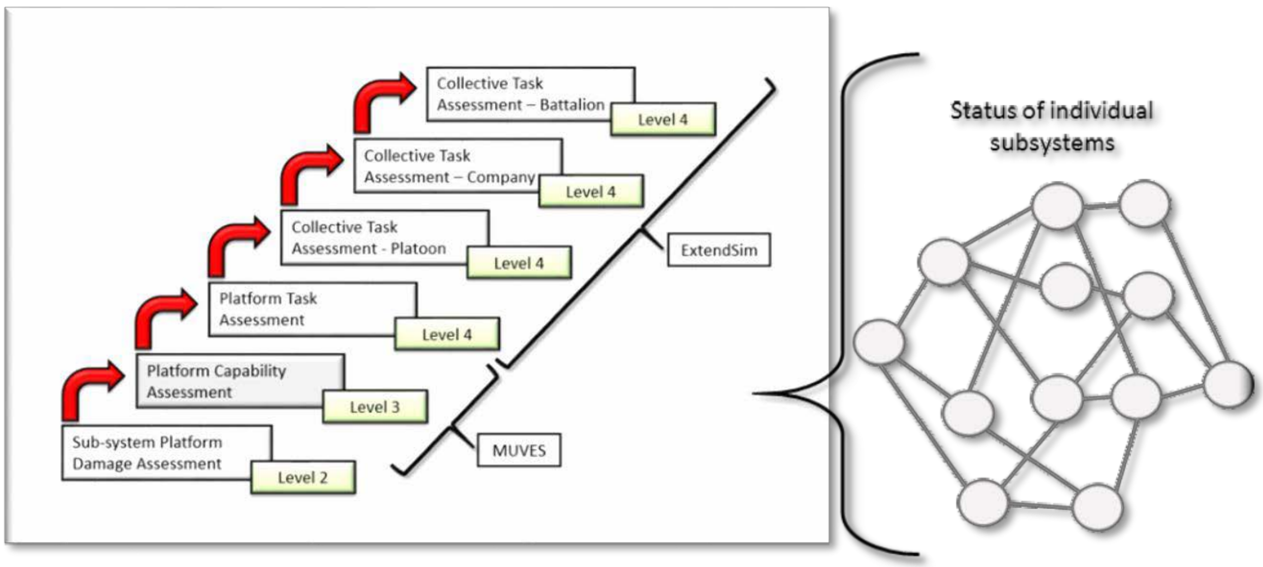


Figure 46 Assessment Logic - Platform Capability

By examining the Reliability and Repair data tables we could determine the specific sub-systems that failed leading to the degraded capability assessments for the two impacted platforms. The MaintFailEventDLLData and Log_MaintRecovery point to a suspension failure that occurred at approximately minute 1572 (simulation time) on RIFLE PLT1 VEH SEC 3_4. For CO_TM A HQS CDR, the log reveals that this platform experienced a transmission failure at a much earlier event prior to P108, at approximately minute 126 simulation time. The Repair logs showed that the failure was not corrected in Assembly Logan thus the platform continued with a degraded capability state.

At the lowest level of the assessment, MUVES determined the impact of each interaction on the components of the platform's subsystem (Figure 47). In the example described, MUVEs determined the individual components that failed that caused the suspension and transmission failures listed above for RIFLE PLT1 VEH SEC 3_4 and CO_TM A HQS CDR respectively.

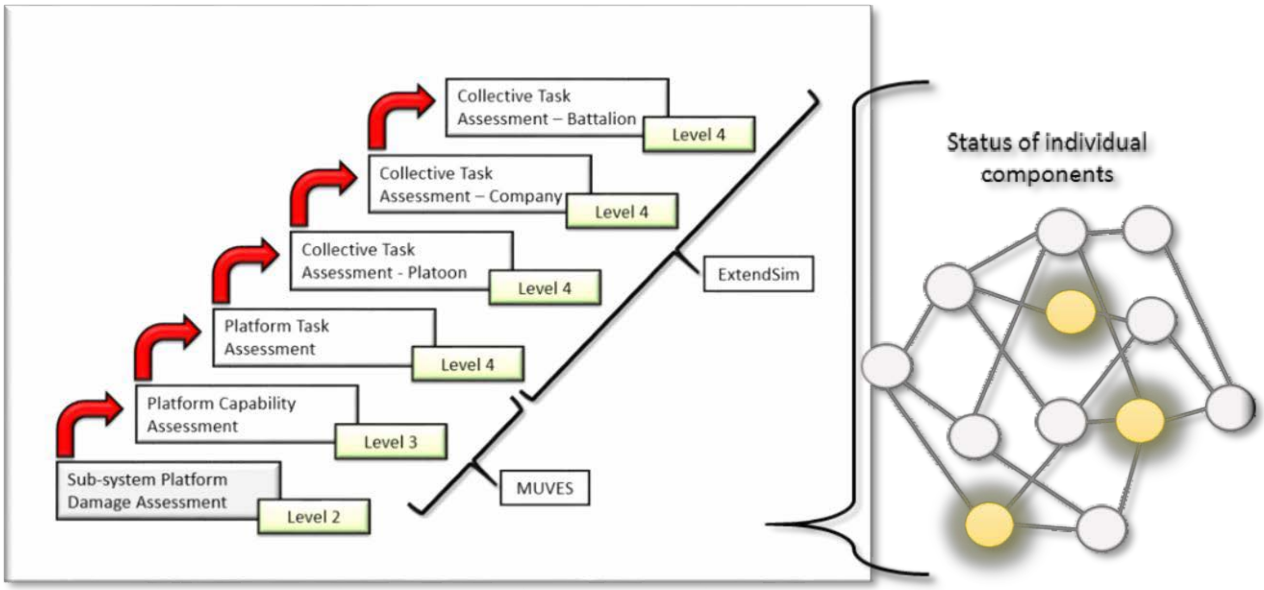


Figure 47 Assessment Logic - Sub System Platform Logic

6 • Findings Conclusions and Lessons Learned

6.1 Findings

a. The concept of using formal task structures to evaluate effectiveness of platforms of interest by applying and integrating the results of kinetic interactions, reliability failures and repair events in real time using dynamic geometry was successfully demonstrated.

b. This was the first time that a dynamic system level survivability model has been linked via an API to pass data back and forth with an operational, FoF type model.

c. The MMF structure facilitated productive collaboration between:

- Operational SME (Mission Specification role)
- Operational model developer (Force on Force model development role)
- System model developer (System vulnerability/survivability model development role)

d. Productive collaboration enabled rapid development of:

- Common metrics
- Functional API between FoF and System models

e. The resulting integrated model produced reasonable, realistic results given notional system data and task assessment logic.

f. Run results were easily transferred to mission thread and applied to analyze and visualize mission impact across multiple echelons.

g. Overall structure enabled drill down from mission impact/collective task failure to component failure and associated cause.

6.2 Conclusions

a. Mission specification (via MDMP analysis and generation of MMF mission specification products) is critical to enable assessment of performance and operational effectiveness using data generated from multiple sources. The formal task construct and especially the mission thread which essentially forms a task-based fault tree for each operation/mission provides a mission structure that can be continuously updated and reassessed as new data from task execution events (e.g., developmental test events, experiments, field data collection from instrumentation, etc.) is added.

b. Collaborative development of common metrics within context of MMF structure is a springboard for Agile development of integrated M&S tool suites from System to Operational level. This has implications for T&E, requirements analysis and integration, system design and systems engineering (MBSE), experimentation and integrated training and readiness assessment.

c. Model architecture and lessons learned should be considered for integration in existing Force on Force models and simulations and development of APIs with system models.

d. Potential for extension of application to assess network performance and effectiveness as a complex System of Systems (and other DOTMLPF solutions) in an operational context should be explored.

6.3 Lessons Learned

a. Time Ordered Event List (TOEL). The time ordered event list (Figure 48) was the primary instrument for describing the events in the operational scenario and translating the intent into language that could be used for input to the simulation model. After experimenting with different formats, we eventually settled on a format with headers of the columns being the event numbers. Underneath each of the event numbers were areas for OWNFOR (Blue) narratives that described both the tasks to be performed, the entities, and the interactions that were to occur, which included details of the OPFOR narrative. Additional rows at the top were included to specify the scenario and simulation time for the events. In the left-hand column, subsequent rows listed every entity modeled in the simulation. Cells to the right of the entity, aligned with each event column, were filled with data about the entities relevant to the event. In most cases, we used those cells to indicate the order of march along routes or axes for entities moving. In some cases, we used those cells to simply indicate which entities were involved in the specified event. In later versions of the TOEL, we added a column that identified the routes or axes that entities followed during specific phases of the operation. This allowed for filtering of the TOEL to depict only those entities we were concerned with as we were modeling the interactions (Figure 48). A copy of the TOEL used for the Deliberate Attack and Exploitation phase may be obtained by contacting the authors.

b. There are some additional changes to the TOEL construct that we recommend which would enhance its usefulness. First, we recommend adding a column to identify which higher level organization each entity was task organized to during specific phases of the operation. This would provide another way to filter information as the model was being developed from the TOEL. Secondly, for each phase of the operation, the events were numbered using the convention P101, P102, P103, etc. Since the scenario had three phases, there were events in each phase named P101, P102, etc. This made it difficult to cleanly translate these event numbers into the simulation. We recommend using a continuous numbering system for the events across the phases. Third, although combining the OPFOR narrative with interaction block worked, separating OPFOR narrative and interaction descriptions into distinct entry fields would have increased clarity.

		Axis sort	Seq. #	P101	P102	P103	P104	P105	P106
			Time	800	800	900	930	1000	1000
			Simulation Time	1470	1470	1530	1560	1590	1590
			Blue Narrative	cross LD and initiate screening along screen line	Initiate movement on Axis Blue	Cross LD and initiate movement along Axis White	Cross LD and initiate movement on Axis Red	Reach fixing positions outside OBJ DAYTON. ABCT TAC CP remains 1km behind 2/7 CAB front line.	ABCT TAC CP initiates scheduled fires on 2 Mech Battalion high value targets.
			Interaction					Elements of 2 Mech Battalion begin engaging with direct fire and mortars)	2 Mech Battalion companies attempting to disengage and fall back to the south
1	Blue Entity	Axis Sor							
2	ABCT BSTB MP TM 2_6	White				3		3	
3	ABCT BSTB MP TM 3_6	LS							
4	ABCT BSTB MP TM 4_6	LS							
5	ABCT BSTB MP TM 5_6	LS							
6	ABCT BSTB MP TM 6_6	LS							
7	CO_TM A HQS 1SG	Red					17		
8	CAB1 MTR SQD 1_4	Red					33		
9	CAB1 MTR SQD 2_4	Red					34		
10	CAB1 MTR SQD 3_4	Red					35		
11	CAB1 MTR SQD 4_4	Red					36		
12	ABCT BSTB CURRENT OPS INTEL	LS							
13	ABCT BSTB CURRENT OPS OPS	LS							
14	ABCT MAIN C4 OPS	LS							

Figure 48 Extract of Time Ordered Event List - Deliberate Attack Exploitation

c. Platform Task and Collective Task Matrices. As explained previously, we used Platform Task and Collective Task matrices to analyze and illustrate the relationships of the tasks to individual entities and entity types. Additionally, we used this series of matrices to analyze and illustrate the relationships of capability states to platform tasks and platform tasks to collective tasks. Task matrices provided an effective means to describe the operators associated with the state changes between the levels of the MMF (e.g. $O_{2,3}$). However, the approach was not efficient and was subject to human error since entries in the MS Excel based spreadsheets were not linked across the multiple worksheets. Furthermore, the matrices were not linked to the MS Project file that contained the mission threads. Future efforts should leverage a database, like MS Access (or linked spreadsheets in MS Excel at a minimum), for the matrices which in turn should be linked to the mission thread files.

d. Platform and Collective Task Assessment blocks. The Platform Task Assessment (Plat_Cap_Assmt) block construct allowed placing the blocks at any location in the model, providing a great deal of flexibility. It did require using customized blocks for the collective task assessment for each event point studied. While a sound approach for the study, the ExtendSim blocks were very large adding significant size to the overall model. Moreover, the structure of the Collective Task Assessment (Col_Cap_Assmt) blocks did not lend themselves to easily expanding the number of examined event points. We recommend redesigning these blocks into two different blocks. The first block would be inserted in the model near the studied event points, as was done in this study, but limiting its functions to reading relevant entity data and writing the updated assessment to the database. The first block would then pass data to a single block to conduct all assessment calculations, both platform and collective. The internal mechanisms of the assessments themselves would remain the same as they are currently constructed. Only the method of passing entities through the assessments would be altered. This would allow for rapidly adding assessments at additional event points and reducing the overall model size.

e. API. Developing and debugging the API was the single most difficult technical challenge during the study. The writing of the initial API code required approximately 80 hours of work divided between developers working both the ExtendSim and MUVES sides of the API. However, during the development and debugging process, only one computer had the entire suite of ExtendSim and MUVES code which extended the testing timelines. We recommend that copies of the complete suite of tools be available to all personnel coding and debugging the API for future efforts.

f. Additional capability descriptors. Additional capability descriptors generated for notional system data in MUVES would have enabled more robust assessment of effects of interactions.

7 • WAY AHEAD

7.1 Near term:

We recommend that the Deputy Undersecretary of the Army for Test and Evaluation (DUSA TE) establish an Integrated Process Team (IPT) with representation from Army Test and Evaluation Command (ATEC), G8 Army Modeling and Simulation Office (AMSO), Training and Doctrine Command (TRADOC) Army Capabilities Integration Center (ARCIC) and Analysis Center (TRAC), Research Development and Engineering Command (RDECOM) and possibly Assistant Secretary of the Army/System of Systems Engineering and Integration (ASA/ALT SOSE&I) to generate recommendations for potential implementation and expansion of study approach for application to network, cyber, sustainment and other areas.

7.2 Medium term:

Pending recommendations from the IPT, we recommend that DUSA TE champion an ATEC – led AND ARCIC – supported effort to convert requirements documents, supporting Capabilities Based Assessment (CBA) documents, and selected TRAC standard scenarios into MMF compliant mission specification products.

7.3 Long term:

We recommend that the Army apply resulting expertise and mission specification results to spur development of lower level ontologies organized under an umbrella MMF – based upper ontology to enable an analytical structure suitable for digesting and converting big data on systems into actionable information to support decision making.

8 - References

1. *Live Fire Testing*. **National Defense Authorization Act for FY 1987, contained in Title 10, United States Code, Chapter 139, Section 2366**. October 18, 1986.

2. C.L. Nail, *Vulnerability Analysis for Surface Targets (VAST) - An Internal Point - burst Vulnerability Assessment Model - Revision 1*. s.l. : **Computer Sciences Corporation, 1982. Technical Manual CSC TR-82-5740.**

3. Paul H. Deitz and Alvars Ozolins. *Computer Simulations of the Abrams Live-Fire Field Testing.* Aberdeen Proving Ground : Ballistic Research Laboratory, 1989. **Ballistic Research Laboratory Memorandum Report BRL-MR-3755**.

4. Paul H. Deitz, *A V/L Taxonomy for Analyzing Ballistic Live-Fire Events.* Aberdeen Proving Ground : U.S. Army Research Laboratory, 1996. **U.S. Army Research Laboratory Technical Report ARL-TR-1274.**

5. Paul H. Deitz and Michael W. Starks, *The Generation, Use, and Misuse of "PKs" in Vulnerability/Lethality Analyses.* No. 1, s.l. : **The Journal of Military Operations Research**, 1999, Vol. 4, pp. 19-33.

6. Jack H. Sheehan, Paul H. Deitz, Britt E. Bray, Bruce A. Harris, Alexander B.H. Wong,*The Military Missions and Means Framework.* Orlando : **Interservice/Industry Training, Simulation, and Education Conference, 2003. I/ITSEC 1257.** p. 14.

7. Beth S.Ward, Paul J. Tanenbaum, Keon U. Burley, Paul H. Deitz, Britt E. Bray, Richard S. Sandmeyer, Jack H. Sheehan, *A Demonstration to Assess Effectiveness, Suitability, and Survivability With the Missions and Means Framework.* Aberdeen Proving Ground, MD : **Army Research Laboratory, 2012. Technical Report. ARL-TR-6271**.

8. Paul H. Deitz, Britt E. Bray, James R. Michaelis, *The Missions & Means Framework as an Ontology.* Baltimore : **Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VII**, 2016. Proceedings of the SPIE. Vol. 9831.

9. Paul H. Deitz, James R. Michaelis, Britt E. Bray, Michael A. Kolodny, *The Missions & Means Framework (MMF) Ontology: Matching Military Assets to Mission Objectives.* London : ICCRTS, 2016. **Proceedings of the 21st International Command and Control Research and Technology Symposium**.

10. Headquarters, Department of the Army. *Concept Development, Capabilities Determination, and Capabilities Integration.* **TRADOC Regulation 71-20.** 2013.

11. Chairman of the Joint Chiefs of Staff. *Joint Capabilities Integration and Development System.* **Chairman of the Joint Chiefs of Staff Instruction 3170.01.** 2015. CJCSI 3170.01.

12. Headquarters, Department of the Army. *Armored Multi-Purpose Vehicle (AMPV)* **Operational Mode Summary / Mission Profile (Draft) Version 1.1**. s.l. : U.S. Army, 2012.

13. Christopher Emond, Matthew Koehler, Sara Kasten, Greg Lee, Jeff Lochow, Bruce Tanner, Cliff Thurman, *Multi-Level Scenario Module 2: IX Corps.* Ft. Leavenworth, KS : TRADOC Analysis Center, 2008. **Operational Scenario Documentation. TRAC-F-TR-08-063**.

14. Headquarters, Department of the Army. **Army Doctrinal Reference Publication (ADRP) 1-03**, *The Army Universal Task List.* Washington : U.S. Army, 2015.

15. Headquarters, Department of the Army. **Training Circular (TC) 3-90.6** *Brigade Combat Team Collective Task Publication.* Washington : U.S. Army, 2011.

16. Headquarters, Department of the Army. **Training Circular (TC) 3-90.5,** *Combined Arms Battalion Collective Task Publication.* Washington : U.S. Army, 2012.

17. Headquarters, Department of the Army. **Army Training Publication (ATP) 3-90.5,** *Combined Arms Battalion.* Washington : U.S. Army, 2016.

18. Headquarters, Department of the Army. **Training Circular (TC) 3-90.1,** *Armor and Mechanized Infantry Company Team Collective Task Publication.* Washington : U.S. Army, 2013.

19. Headquarters, Department of the Army. **Army Training Publication (ATP) 3-90.1,** *Armor and Mechanized Infantry Company Team.* Washington : U.S. Army, 2016.

20. Headquarters, Department of the Army. **Training Circular (TC) 3-21.8**, *Infantry Rifle and Mechanized Platoon Collective Task Publication.* Washington : U.S. Army, 2013.

21. Headquarters, Department of the Army. **Army Training Publication (ATP) 3-21.8,** *Infantry Platoon and Squad.* Washington : U.S. Army, 2016.

22. Headquarters, Department of the Army. **Training Circular (TC) 3-20.15,** *Tank Platoon Collective Task Publication.* Washington : U.S. Army, 2013.

23. Headquarters, Department of the Army. **Army Training Publication (ATP) 3-20.15,** *Tank Platoon.* Washington : U.S. Army, 2012.

24. Headquarters Department of the Army. **Training Circular (TC) 3-20.98,** *Reconnaissance Platoon Collective Task Publication.* Washington : U.S. Army, 2013.

25. Headquarters, Department of the Army. **Army Training Publication (ATP) 3-20.98,** *Reconnaissance Platoon.* Washington : U.S. Army, 2013.

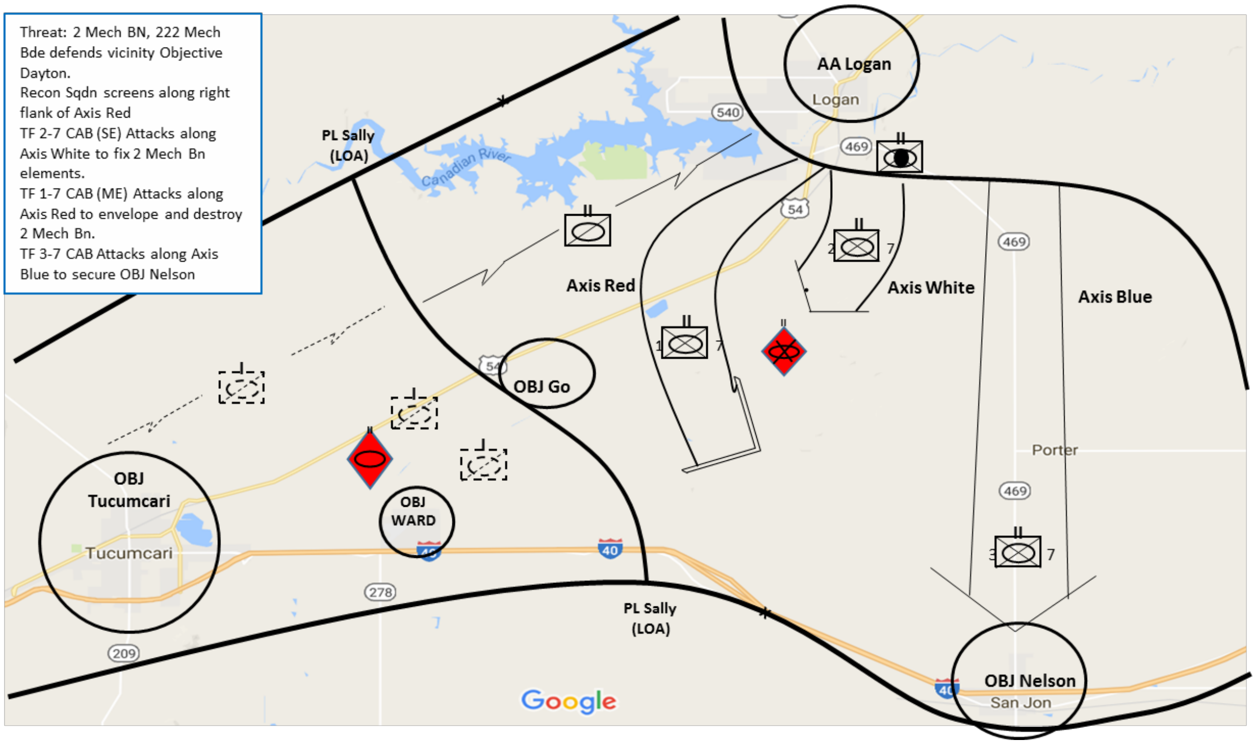
26. Headquarters, Department of the Army. **Training Circular (TC) 3-21.90,** *Mortar Platoon Collective Task Publication.* Washington : U.S. Army, 2013.

27. Headquarters, Department of the Army. **Field Manual (FM) 3-96,** *Brigade Combat Team.* Washington : U.S. Army, 2015.

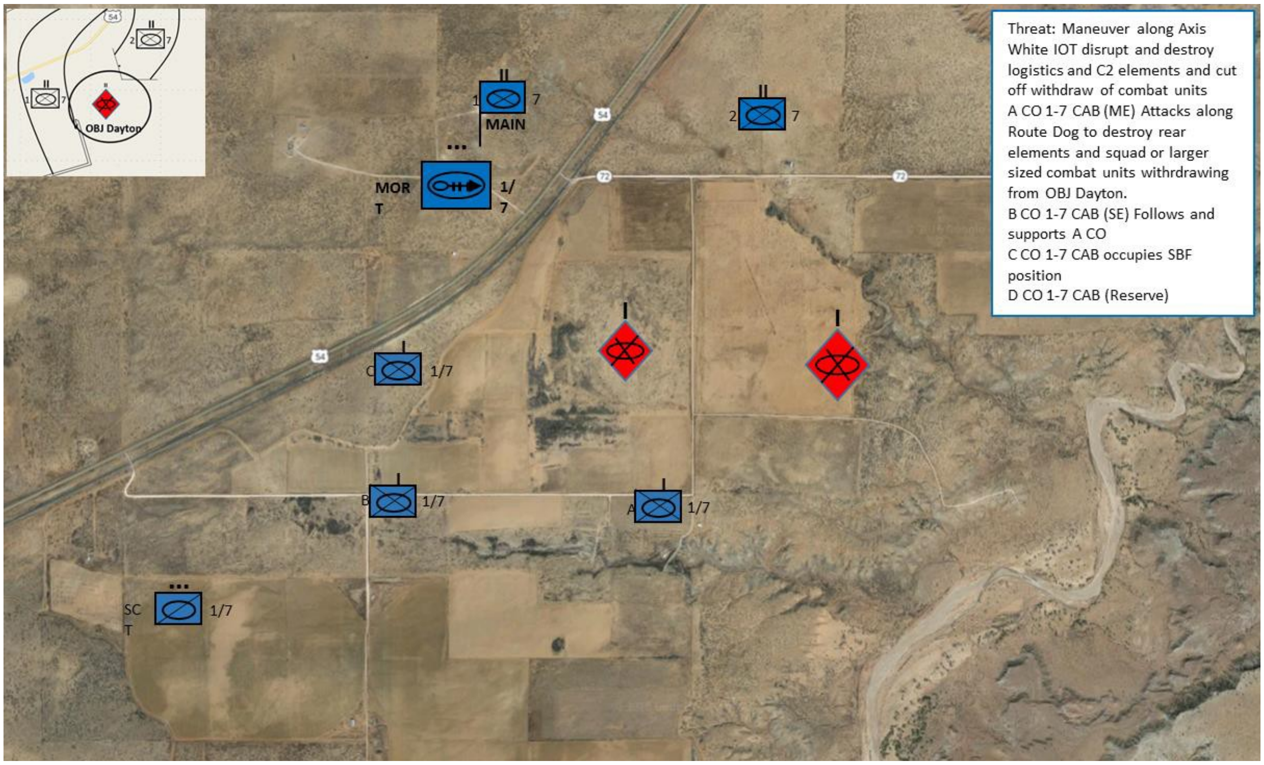
28. Mounted Requirements Division, Maneuver Center of Excellence. *ARMOR/CAVALRY REFERENCE DATA BRIGADE COMBAT TEAMS*. **Ft. Knox Supplemental Manual (FKSM) 71-8.** Ft. Knox, KY, USA : U.S. Army, May 2011.

APPENDIX A - MDMP PRODUCTS

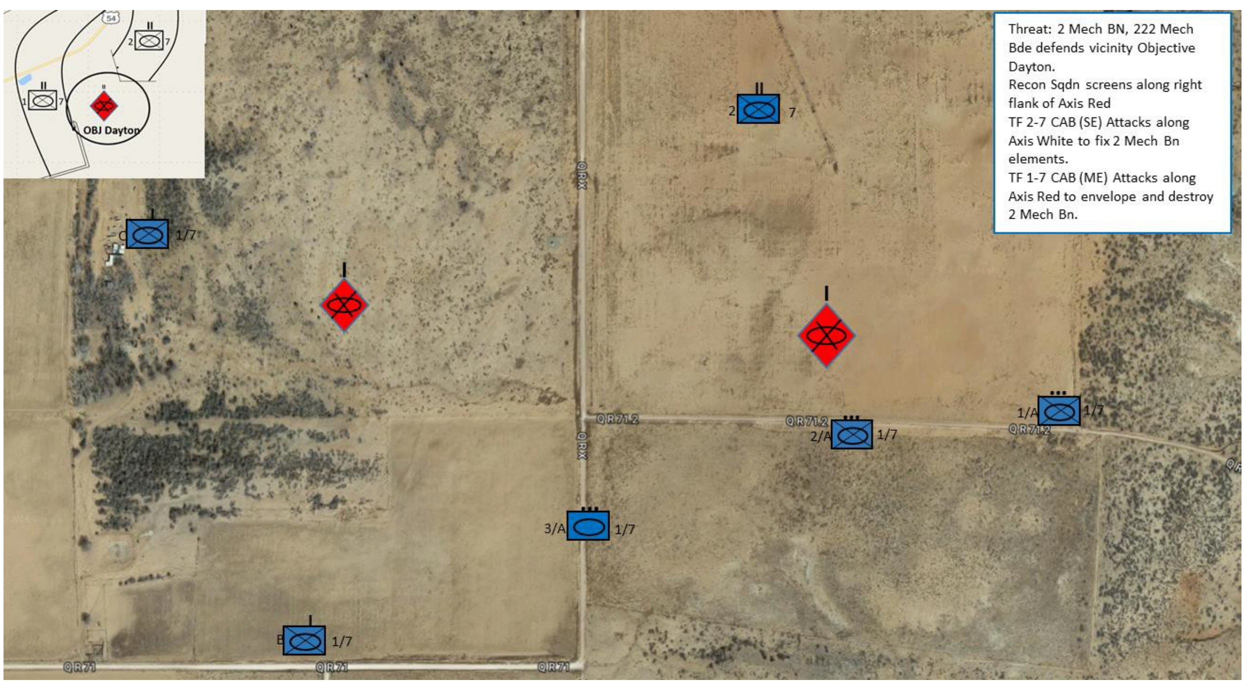
7th ABCT Deliberate Attack & Exploitation



Course of Action Sketch and Statement for ABCT – level Deliberate Attack and Exploitation.



Course of Action Sketch and Statement for CAB-level Attack on Objective Dayton.



Course of Action Sketch and Statement for CO/TM-level Attack on Objective Dayton.

APPENDIX B - MODEL DESIGN AND DEVELOPMENT

B.1 Development of the Model.

B.1.1 Model Layout

The overall model file was laid out linearly from left to right by phase of the TOEL. The screen shot in Figure 49, below displays a high-level view of the Graphic User Interface (GUI) for the model. The simulation is initiated with all model entities located in Assembly Area (AA) Boise, the final assembly area prior to commencing the extended tactical road march phase of the derived operational scenario. The high-level view is divided into three segments delineated by dotted red lines. The left most segment consisted of:

AA Boise and the set of modeling blocks and sub-blocks that composed it.

Phase 1 – Extended Tactical Road March and Assembly Area (AA) Logan.

The center segment consisted of Phase 2 – Deliberate Attack and Exploitation.

The right most segment consisted of Phase 3 – Deliberate Attack Urban Environment

Continue to Section B.1.2 for a detailed explanation of the modeling constructs and logic used to design and build each of the three segments.

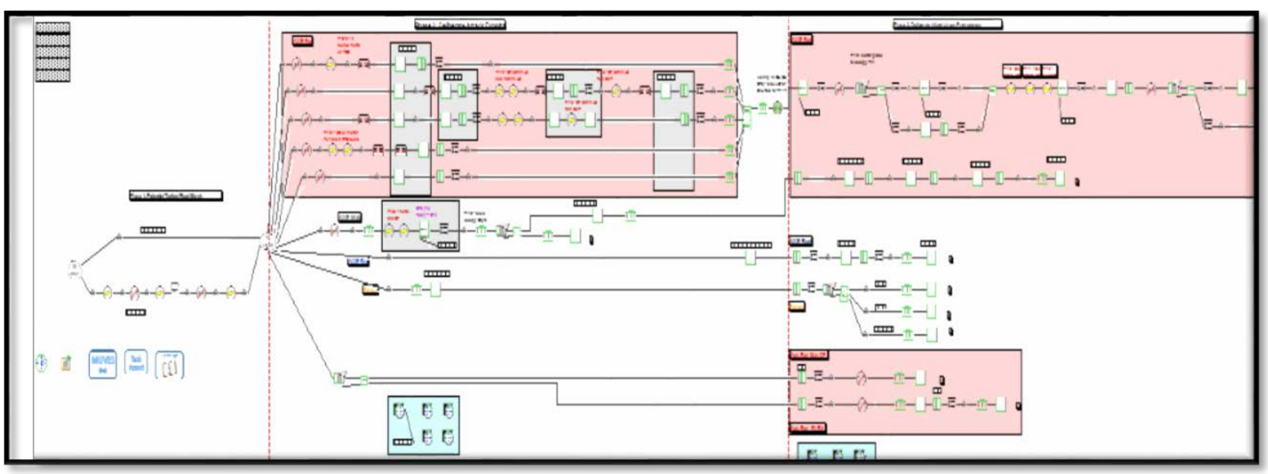


Figure 49 Screen Shot of Model Layout

Figure 50 shows a blow-up version of the Control Panel, located in the upper left portion of the model space. The Control Panel consisted of five different buttons that enabled the user to run the simulation, open the database, look or add items to the notebook, and turn on and off the animation in the model.

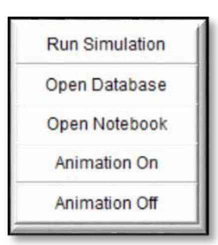


Figure 50 Control Panel

The Executive block, located in the lower left hand corner controlled the overall execution of the simulation. Next to it was the Data Init block used to reset data for the number of kilometers driven by individual platforms at the beginning of each replication. Next to it was the MUVES Manager block which controlled ExtendSim's interface with the API connecting ExtendSim to MUVES. Next to it was the Task Assmt block that contained the collective task assessment constructs and blocks that exported data to MS Excel for the collective and platform task assessments. Next to the Task Assmt block was the Event Logs block that exported to MS Excel log data associated with the ballistic, reliability, and repair events in the simulation. Figure 51 provides a blow-up display of these blocks.

Figure 51 Executive and Selected Component Blocks

B.1.2 Key Modeling Constructs and Logic. The final model was constructed with one database consisting of 40 separate data tables. Each data table had a unique name as well as a number designation which was identified as a number in brackets. For example, for table Entity_Data [17], the name of the table was Entity_Data and the number designation was 17. The ExtendSim blocks described in the remainder of this section referenced the individual data tables using the number designation. Each data table is described in greater detail in section B.2

B.1.2.1 AA Boise.

Throughout the model, we used hierarchical blocks (H-blocks) to contain a series of sub-models that kept the model space relatively clean. The first of these H-blocks was AA Boise which consisted of nested H-blocks – H-blocks within H-blocks. In Figure 52 below, the top most window is the top-level H-block of AA Boise; highlighted in black in the top H-block is a nested sub-model which is opened below it, and a further sub-model that is opened below it. In AA Boise, all entities were created and assigned some initial attributes including name, type, the number of operational kilometers that each entity had driven prior to the beginning of the simulation (used for reliability modeling), and the number of kilometers the entity would travel prior to it encountering a reliability failure. The entities departed AA Boise based on the SP Time in the Entity_Data [17] table based on the TOEL. Based on the individual entities' missions, they departed AA Boise following either Route Dodge or the route for the screening force.

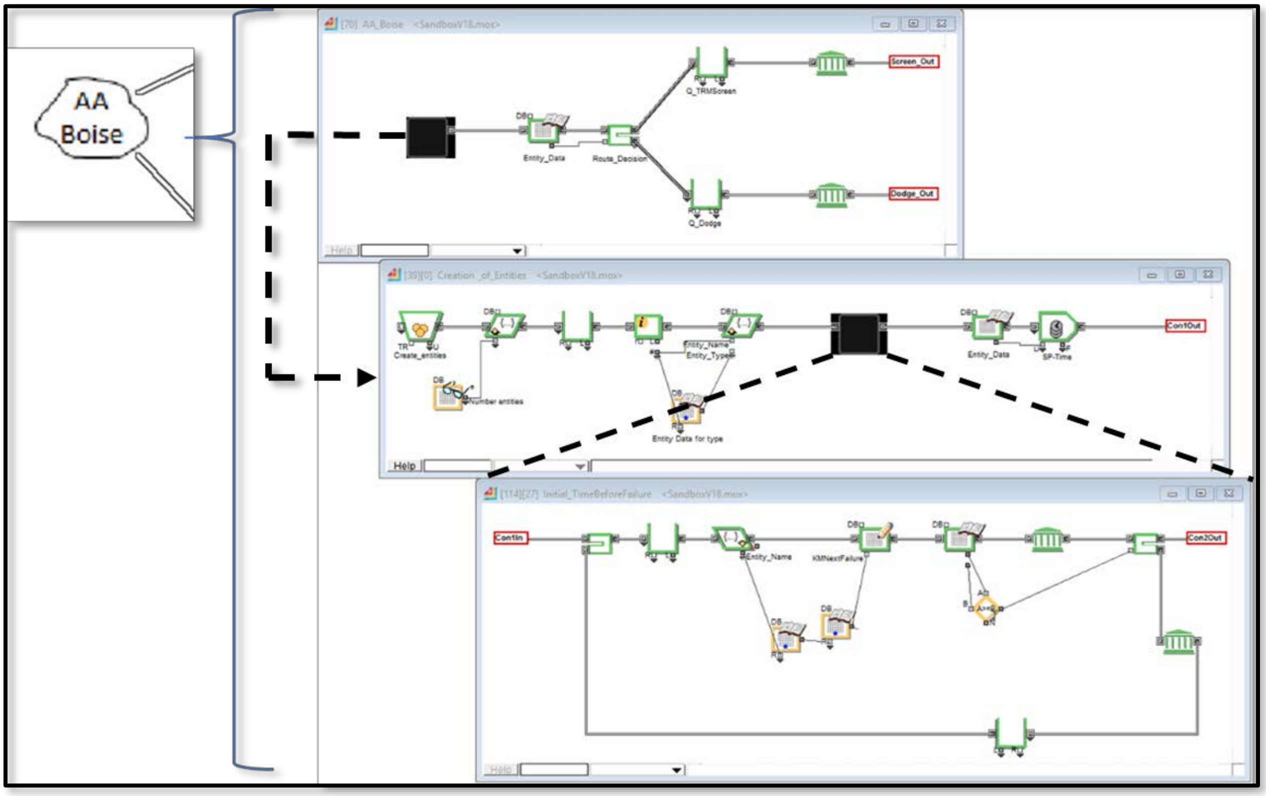


Figure 52 Screen Shot of H-block Decomposition of AA Boise

B.1.2.2 gment_transport.

After moving through the AA Boise H-, entities passed through the first of many Segment_transport blocks in the model (Figure 53). Segment_transport blocks were one of several custom blocks developed for this study that were placed in the FoFconstructs.lix ExtendSim library. The Segment_transport blocks controlled the rate of movement of all entities through all portions of the model. Each route segment was identified using the drop-down menu in the upper left hand corner of the block window. To control movement, the block read the Route_Segment_Data [22] table to obtain the length of the route segment and speed along which each entity would travel and then calculated the time of travel for the entity along the route segment. Additionally, it updated the Entity_Status [19] table to update the number of kilometers the entity travelled by the length of the route segment.

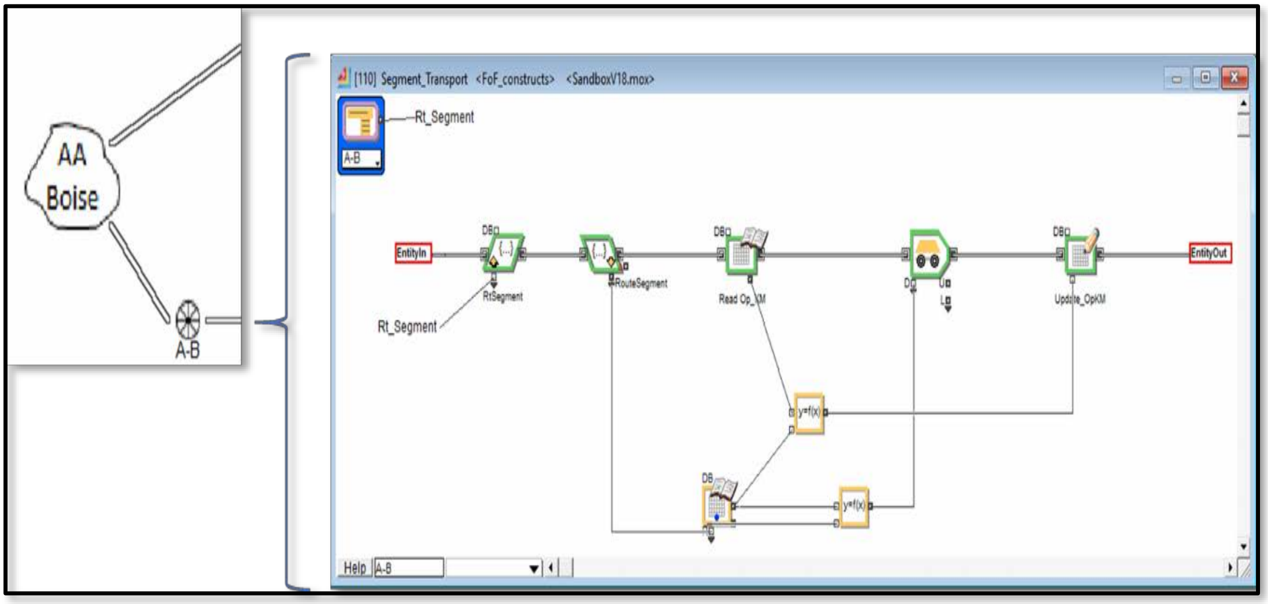


Figure 53 Screen Shot of Segment_Transport Block

B.1.2.3 Kinetic_Event (Ballistic Event).

All ballistic interactions were controlled by the Kinetic_event blocks (Figure 54), also included in the ExtendSim library FoFconstructs.lix. The block first determined if a given entity would be involved in a ballistic interaction at that event point by referencing the probability of entity involvement in a ballistic interaction found in the Entity_Data [17] table. The event point for the interaction was designated by the drop-down menu in the upper left hand portion of the block. If the entity was not involved in an interaction, it simply exited the ballistic event block. If it was involved in the interaction, the block would create an event identification number, EventID, as a reference number, and read the ballistic event parameters from the database. The EventID and ballistic event parameters were passed by the Ballistic Event block to the MUVESManager block which in turn passed them through the API to MUVES. After MUVES calculated resulting capability assessment and passed the data back to ExtendSim, the block calculated the time required to recover the platform if the platform was at a degraded Mobility state.

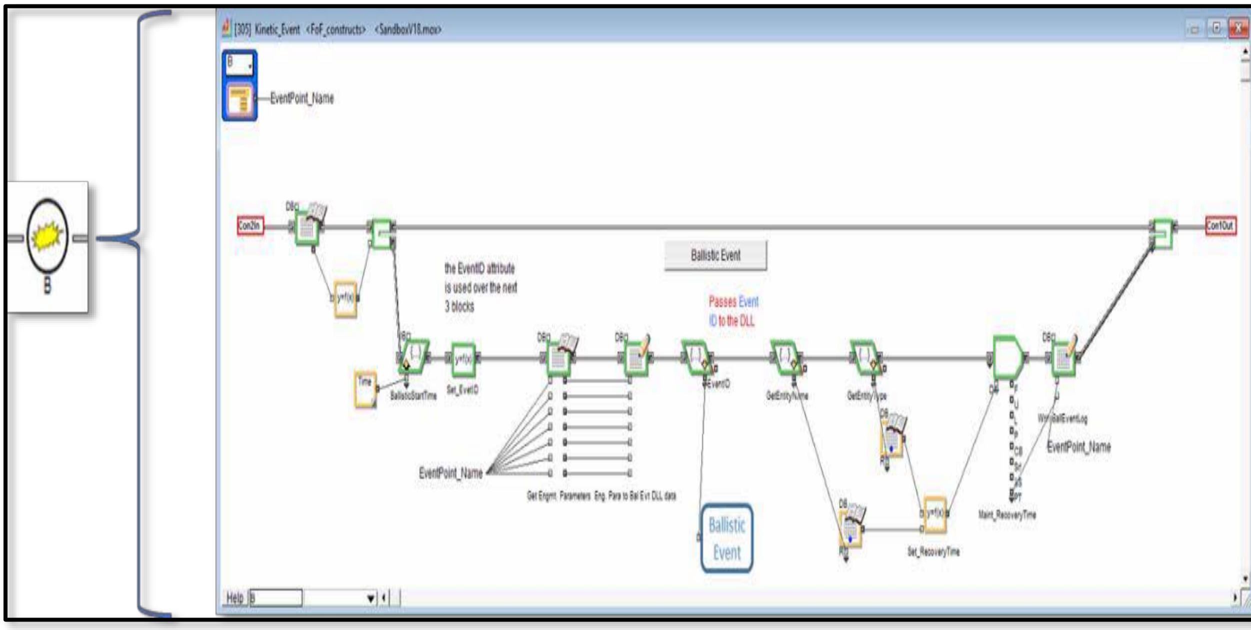


Figure 54 Screen Shot of Kinetic_Event Block

B.1.2.4 aint_Event (Reliability Event).

All reliability failures of the platforms were handled by the Maint_Event blocks (Figure 55), also included in the FoFconstructs.lix ExtendSim library. As with Kinetic_Event block, the maintenance event point for the interaction (failure) was designated by the drop-down menu in the upper left hand portion of the block. By comparing the number of kilometers the platform travelled to the number of kilometers prior to the next failure - both found in the Entity_Status [19] table - the block determined if the platform would experience a reliability failure. If it did not, the platform would exit the block. If platform experienced a failure, the block would create an event identification number, EventID, and along with the entity name, would pass it to the Maintenance Failure block which would in turn pass it to the MUVESManager block. The MUVESManager passed the data through the API to MUVES. After MUVES calculated resulting capability assessment and passed the data back to ExtendSim, the block calculated the time required to recover the platform if the platform was at a degraded Mobility state.

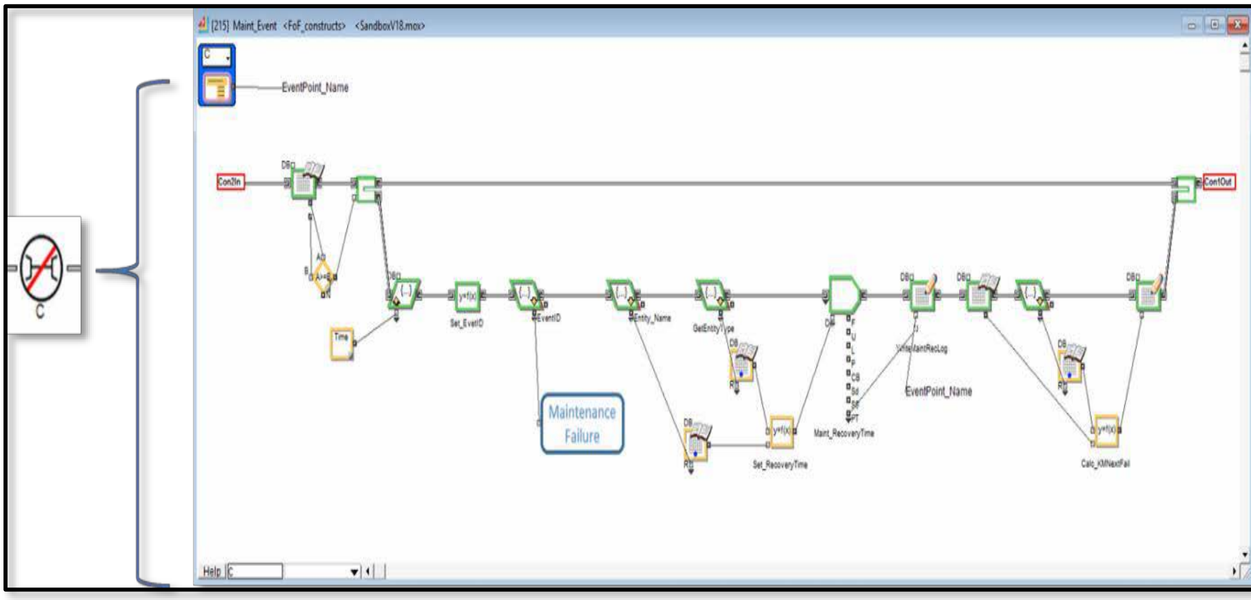


Figure 55 Screen Shot of Maint_Event Block

B.1.2.5 AA Logan.

Like AA Boise, AA Logan was an H-block that contains additional sub-models. Since many of the blocks and constructs in the AA Logan appeared throughout the remainder of the model, they will be discussed separately below.

B.1.2.6 Repair_Event.

As with other interaction blocks, the repair event point (Figure 56) for the interaction was designated by the drop down menu in the upper left hand portion of the block. Since only platform entities could experience either reliability failures or ballistic interactions, non-platform entities immediately exited the block. The block first determined if there was any degradation to a platform's capability state. If not, the platform would exit the block. If so, the block would pass an EventID and entity name through the Repair Event block which would in turn pass it to the MUVESManager block. The MUVESManager passed the data through the API to MUVES. MUVES reset all capability state levels for the entity to 1 and passed the resulting data back to ExtendSim. Using the mean time to repair found in database table Entity_Type_Data [16], the block processed the item for repairs.

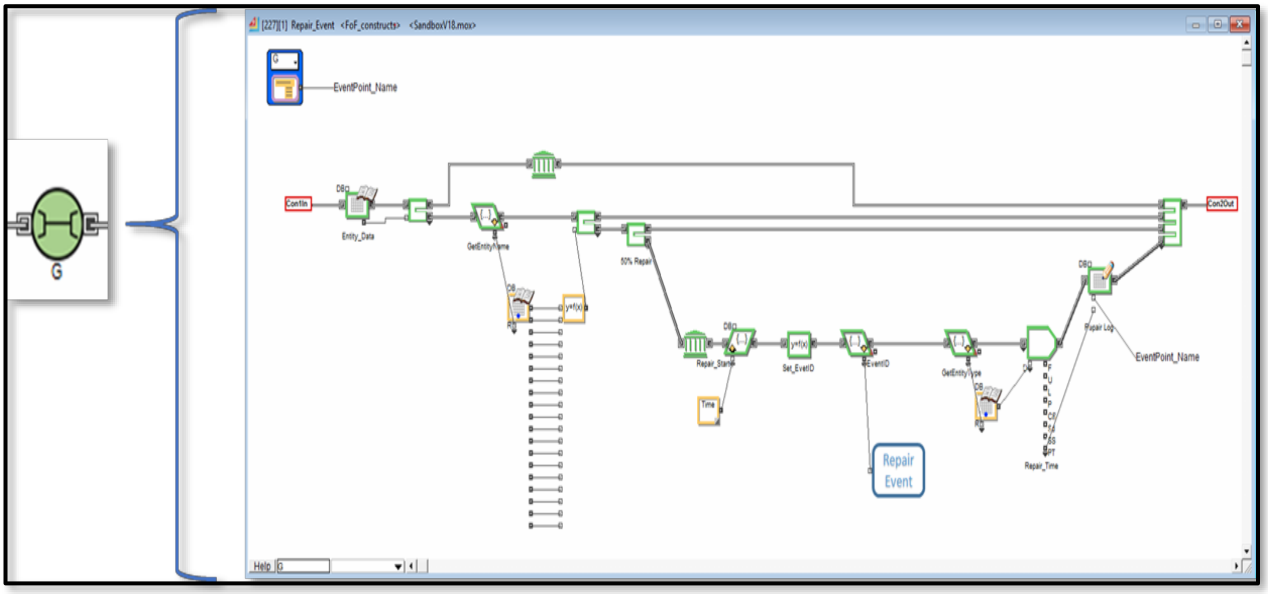


Figure 56 Screen Shot of Repair_Event

B.1.2.7 Gates and Shifts.

Throughout the simulation there were two ways in which we controlled the departure of entities along a route to correspond with the TOEL. The first method employed the use of Gates and Shifts (Figure 57). An individual Gate would either be opened or closed based on the timing specified by the associated Shift. In our simulation, we used this method if all entities on a given route would proceed further in the same order that they arrived at the Gate, in other words, the march order did not change.

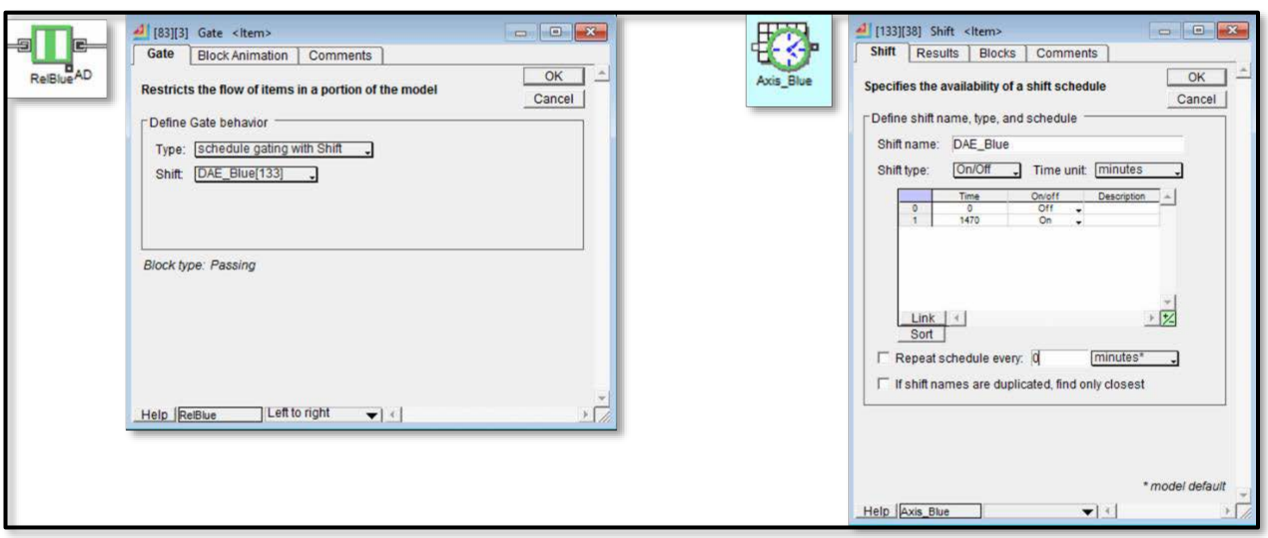


Figure 57 Screen Shots of Example Gate with Associated Shift

B.1.2.8 Equation blocks and Shifts. If the march order of entities beyond a certain point needed to be changed, we used the second method to control the departure of entities along a route: An Equation block and a Shift (Figure 58). The Shift was used in the same way as it was with the Gate and Shift method. In this case, the Equation block accessed the Entity_Data table [17] and field associated with march order sequence for that point in the scenario. Entities were then released from the Equation block following the prescribed march order.

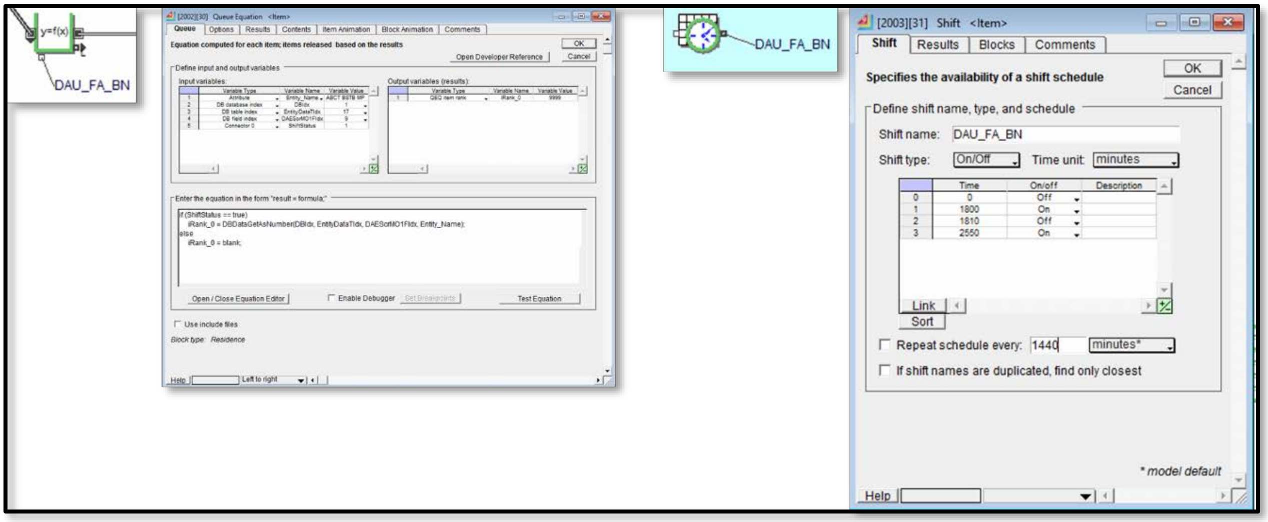


Figure 58 Screen Shots of Example Equation Block with Associated Shift

B.1.2.9 archOrderSpacing.

This simple block (Figure 59), also included in the FOFconstructs.lix library, followed every Gate or Equation block used to control entity release along a route segment. This block delayed every item three seconds to provide spacing between the entities along the route. Using a custom block in the library as opposed to the standard process block enabled the march order spacing to be adjusted for all the MarchOrderSpacing blocks in the model to a standard value at the same time.

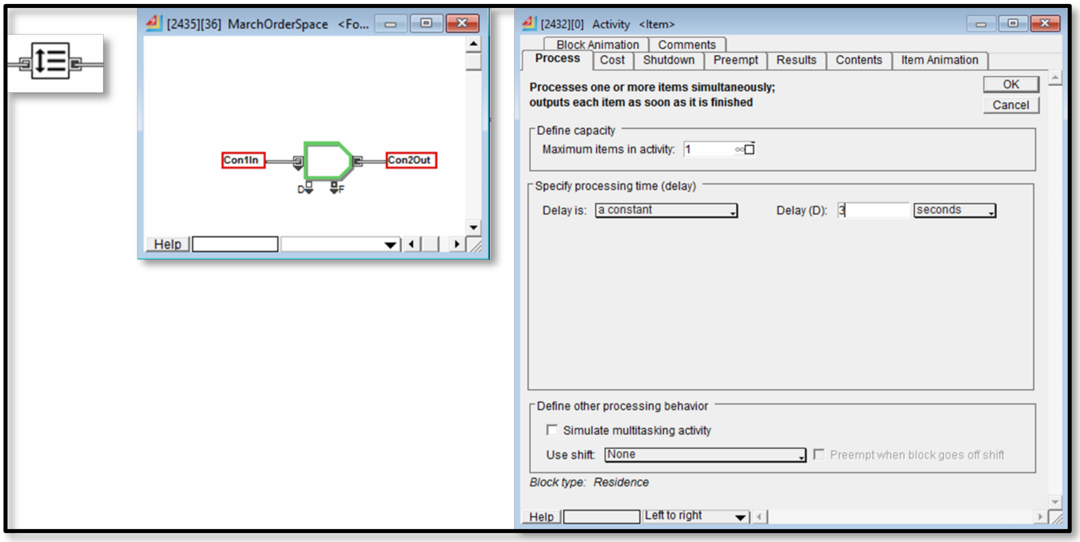


Figure 59 Screen Shot of MarchOrderSpacing Block

B.1.2.10 Platform Task Assessment.

The task assessment for each platform was conducted by the Plat_Task_Assmt blocks (Figure 60), also included in the FOFconstructs.lix library. These blocks were placed at selected points in the simulation immediately prior to where platforms would be required to perform a specific task. The event (P104, for example) was entered in the Platform Event Table entry field (center section of Figure 60). As entities entered the block they were sorted by Platform Type as described in the Entity_Data [17] table (center section of Figure 60). Since not all platforms performed all tasks, only those tasks performed by the platform were assessed (right section of Figure 60). For each platform type, the current capability state was read from the Capability_Assessment [29] table; those capabilities relevant to the task were then entered in an Equation block that calculated the Platform Task Assessment; see Section 5, Model Execution and Results, for a description of the methodology used. The block then wrote the assessment to the appropriate TaskCapPlat_#### table in the database as specified by the Platform Event Table entry field.

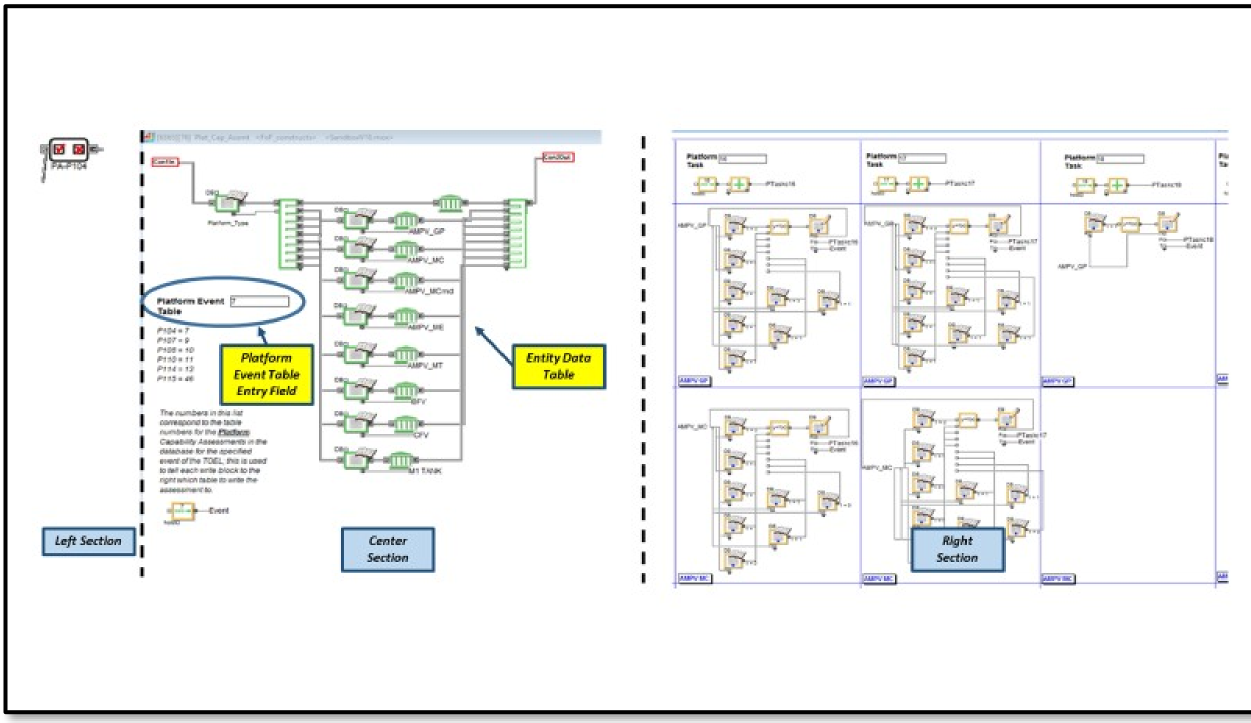


Figure 60 Screen Shot of Plat_Task_Assmt block - extract

B.1.2.11 Collective Task Assessment.

The collective task assessment process was executed by separate H-block constructs within the Task Assmt block located in the lower left hand portion of the model window (See Figures 49 and 51), one for each studied event (P104, P107, P108, P110, P114, or P115). The Task Assmt block also contained 12 Data Import Export blocks, two for each studied event, that exported the data from the platform and collective tasks to MS Excel files after each replication (Figure 61)

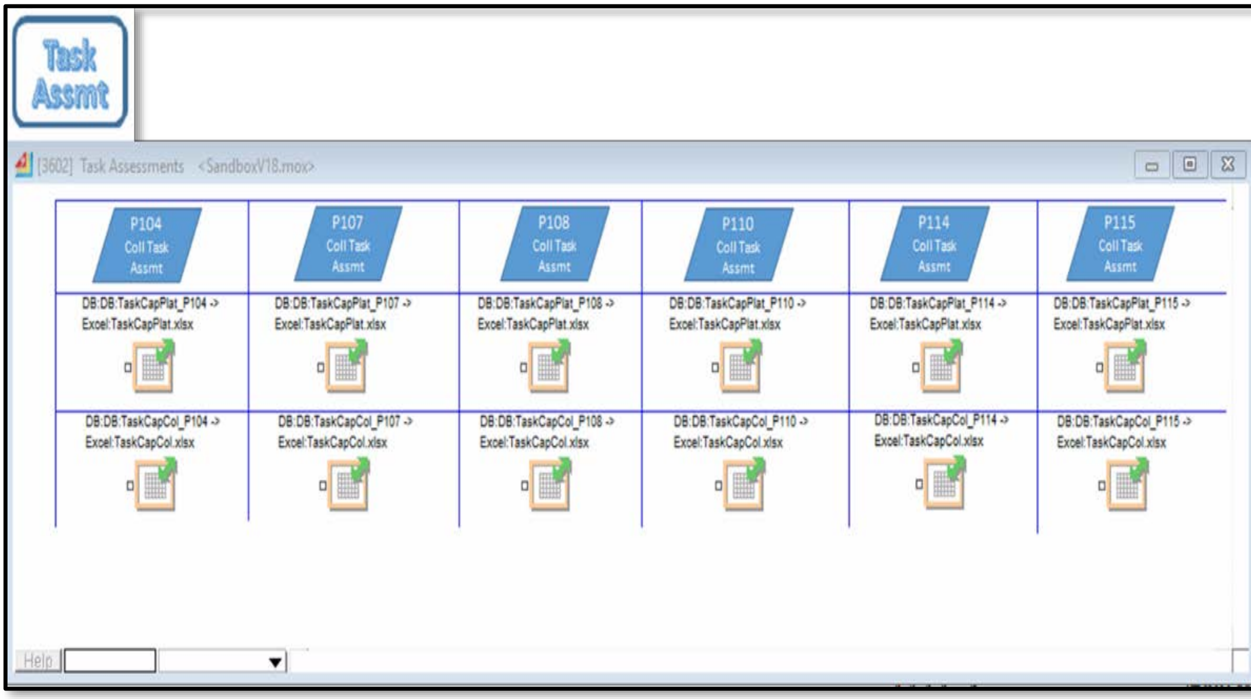


Figure 61 Screen Shot of Task Assmt Block

Like the Platform Task Assessment blocks, these blocks assessed tasks at the same events in the simulation (P104, P107, P108, P110, P114, or P115). Table numbers corresponding to selected events were entered in the Event Table entry field (Figure 62). Instead of reading the Capability_Assessment [29] table, the assessment blocks read the appropriate TaskCapPlat_#### table in the database for the event. An Equation block then calculated the Collective Task Assessment. The assessment blocks then wrote the assessments to the appropriate TaskCapCol_#### table in the database. Underneath the Event Table entry field there was a section that identified when data was to be read for the various calculations. These were near the end of the simulation time, spaced apart by a minute. That section and the associated spacing allowed for the first set of calculations to be done for collective tasks composed of platform tasks. When applicable, the subsequent database reads allowed collective tasks to be calculated using lower level collective task assessments, each built on the level below it; see Section 5, Model Execution and Results, for a description of the methodology used.

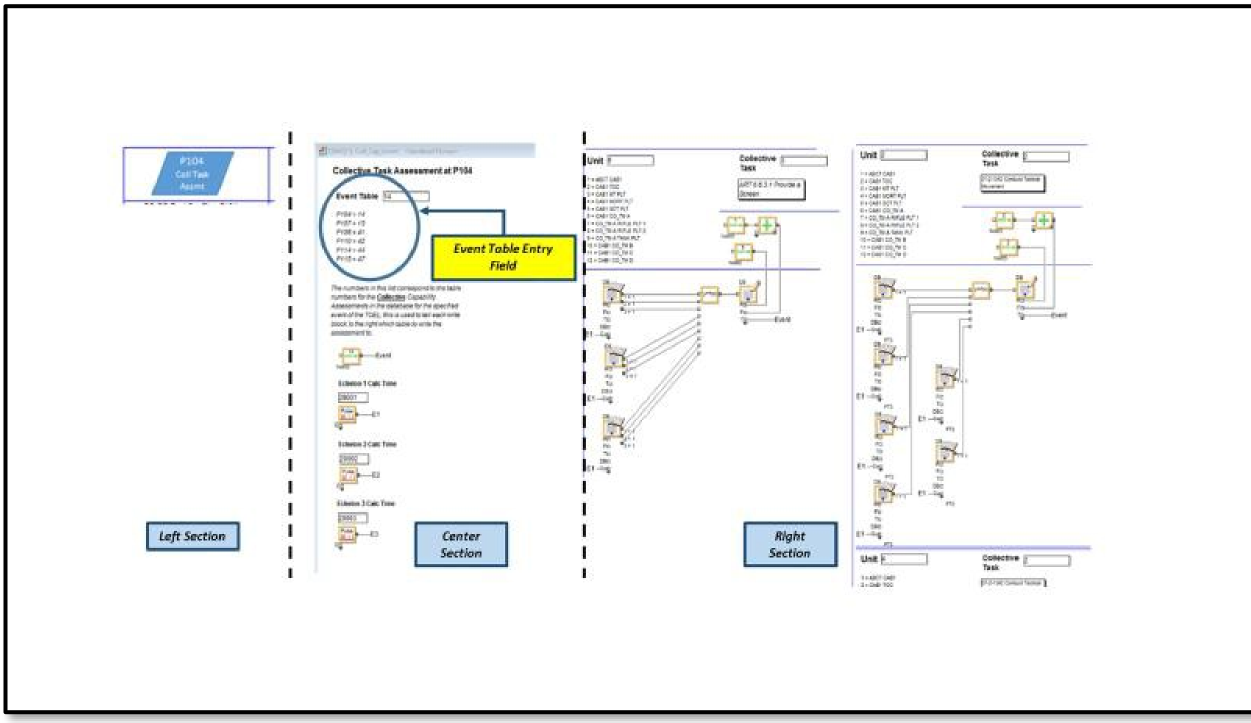


Figure 62 Screen Shot of Collective Task Assessment Block - extract

B.1.2.12 Event Logs.

The Event Logs block near the bottom of the model space (see Figures 49 and 51) contained six Data Import Export blocks. At the end of a simulation replication, these blocks exported the following tables to MS Excel: BallisticEventDLLData [4], MaintFailEventDLLData [28], RepairEventDLLData [30], Log_MaintRecovery [25], Log_BallisticEvent [27], and Log_RepairEvent [32]. For illustration purposes, Figure 63 below includes a depiction of the Data Import Export block opened for the MaintFailEventDLLData [28].

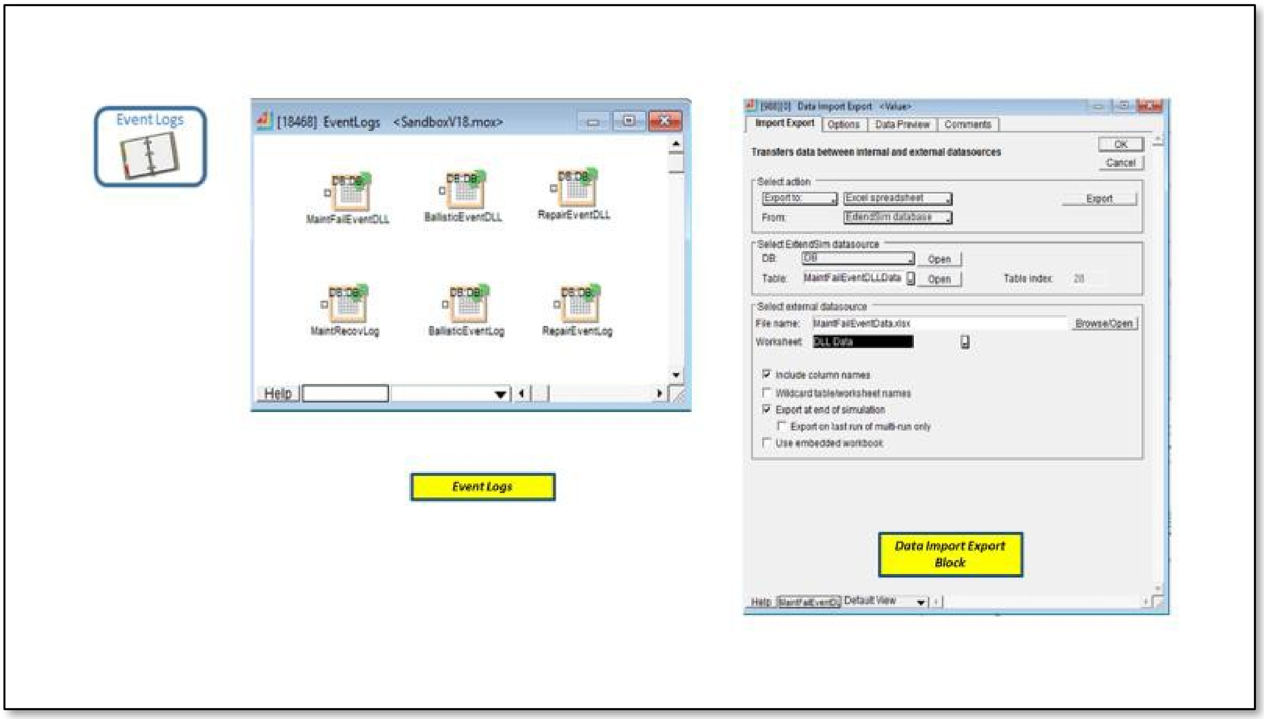


Figure 63 Screen Shot of Event Logs_Data Import_Export Block

B.1.2.13 API Coding Blocks.

Mentioned above, there were four custom blocks, included in the FOF_constructs.lix library, developed to enable ExtendSim to interface with the API: BallisticEvent, MaintFailureEvent, RepairEvent, and MUVESManager. The BallisticEvent, MaintFailureEvent, and RepairEvent were all incorporated into their respective event blocks: Kinetic_Event (ballistic interactions), Maint_Event (reliability events), and Repair_Event. The MUVESManager block was located at the bottom of the model space.

B.2 Data Structure. The final model was constructed with one database consisting of 40 separate data tables. The data tables were organized by tabs into seven different groups of tables: Lists, Input Tables, Output Tables, Status, Location_Data, Log, and DLL. Each data table had a unique name as well as a number designation which was identified as a number in brackets. For example, the name of the Input Table, Entity_Data [17], was Entity_Data and the number designation was 17. The ExtendSim blocks referenced the individual data tables using the number designation. Field names also had numerical designations in addition to a name. Table numbers are included in the tables names throughout the remainder of this section but field numbers are not.

B.2.1 Lists

The list tables all had a simple two column construction consisting of a record number and data column. These tables provided common reference names for frequently used data fields throughout the model and were linked as a "parent" to one or more data tables within the database. The use of parent-child relationships in the database restricted the entries in child table fields to those in the parent table field.

B.2.1.1 List_Cav_Mission [35].

This table contained one field named DAEMsn2. The entries in the table were the second set of missions that platforms associated with cavalry units conducted during the Deliberate Attack Exploitation phase of the operation. This table was the parent for the entries in the field of the same name in the Entity_Data [17] table.

B.2.1.2 List_Entity_Category [2].

This table contained one field named Entity_Category. The first 59 entities in the simulation were designated as vehicles and were played by both ExtendSim and MUVES. The remaining 30 entities were designated as units and were only played by ExtendSim. This table was the parent for the entries in fields of the same name in the both the Entity_Data [17] and Entity_Type_Data [16] tables.

B.2.1.3 List_Entity_Name [3].

This table contained one field named Entity_Name containing the list of names for all entities in the simulation, vehicles and units. This table was the parent for entries in the field of the same name in multiple tables in the database.

B.2.1.4 List_Entity_Name_short [37].

This table contained one field also named Entity_Name containing the list of the names for the first 59 entities in the simulation, those representing platforms and being played both by ExtendSim and MUVES. The Entity_Name field in this table was a "child" of the field of the same name in the List_Entity_Name [3] table. This field was the parent for entries in the field of the same name in the Capability_Assessment [29] table. Although redundant, this table was necessary to restrict the passage of data to that for entities designated as vehicles (platforms) between ExtendSim and MUVES.

B.2.1.5 List_Entity_Type [1].

This table contained one field named Entity_Type containing the list of possible entity types. The entity types included a listing of all possible vehicle (platform) types and unit types for non-vehicle entities. This table was the parent for entries in fields of the same name in several tables in the database including the Entity_Data [17] and Entity_Type_Data [16] tables as well as multiple logs.

B.2.1.6 List_EventPoint_Name [20].

This table contained one field named EventPoint_Name containing the list of points designated in the simulation for events to occur such as ballistic interactions, reliability failures, or repair events. For reference purposes, only, a second column in the list was used to provide some descriptive data about the event point. This table was the parent for entries in fields of the same name in multiple tables in the database, principally the log and DLL data tables.

B.2.1.7 List_Fail_Types [34].

This table contained one field named Fail_Types containing the names of the 10 different sub-systems that could fail on a platform in the MUVES portion of the model. This table was the parent for entries in the field of the same name in the MaintFailEventDLLData [28] table.

B.2.1.8 List_Loiter_Loc [26].

This table contained one field named Loiter_Loc that contained the name of the temporary command post location during the Extended Tactical Road March phase. This table was the parent for entries in the field of the same name in the Entity_Data [17] table.

B.2.1.9 List_Platform_Type [45].

This table was like the List_Entity_Type [1] in that it contained a list of the possible vehicle (platform) types but had one entry, Unit_N, that was used for all non-platform entities. This table was the parent for entries in the field of the same name in the Entity_Data [17] table.

B.2.1.10 List_Routes [18].

This table was a unique list in that it had five different fields applicable to different phases of the operations:

- Route_TRM. This field listed the two possible routes for the Extended Tactical Road March phase, a route for cavalry units conducting the screen and Route Dodge for the main body.
- Axis_DA_Exp. This field six possible routes for the entities to take during the Deliberate Attack Exploitation phase.
- AxisDAU_RedPA1. This field listed two options for entities travelling on Axis Red during the Deliberate Attack Urban Environment phase: one to stop at Position Area 1 and a second to continue without stopping.
- AxisDAURed_End. This field listed two potential branches for entities travelling on Axis Red during the Deliberate Attack Urban Environment phase.
- DAURedDaytonRel. This field listed two potential branches for entities in the vicinity of Objective Dayton on Axis Red during the Deliberate Attack Urban Environment phase.

Each of the fields listed above were the parents for entries in fields of the same name in the Entity_Data [17] table.

B.2.1.11 List_RouteSegment [23].

This table had one field, RouteSegment, that identified individual route segments throughout the simulation designated by the beginning and end EventPoint_Name separated by a dash; for example, Route Segment H-M spanned the distance between event points H and K. This table was the parent for entries in the field of the same name in the RouteSegment_Data [22].

B.2.1.12 List_TaskOrg1-7 [36].

This table had one field, TaskOrg1-7, that identified the teams each entity of 1-7 CAB could be task organized to during the Deliberate Attack Exploitation phase. This table was the parent for entries in the field of the same name in the Entity_Data [17] table.

B.2.1.13 List_Unit_Name [6].

This table had one field, Unit_Name, that listed the names of the units used for collective task assessments. This table was the parent for entries in the field of the same name in the TaskCapCol tables for each event point where assessment occurred.

B.2.2 Input Tables

The Input Tables were the principle tables where data concerning entity and process behavior was entered into the model. The use of the database for this purpose enabled rapid changing of parameters that altered the conditions surrounding and sequencing of events as well as the actions of individual entities.

B.2.2.1 BallisticEventParameterData [31].

This table had eight fields that identified the event point associated with a specific interaction and the parameters associated with ballistic events; the field EventPoint_Name was a child of the same field in the List_EventPoint_Name [20] table. The parameters were established in the ExtendSim database and passed to MUVES through the PI for each ballistic interaction. The following fields contained data for each relevant event point:

- MunitionType. Three munition types were used for ballistic interactions, designated as 1, 2, or 3:
  1. Rocket Propelled Grenade
  2. High explosive – direct fire
  3. Indirect fire
- Range. This field contained the range in meters of the intended target from the weapon firing the ballistic round. This was modeled using a triangular distribution to allow for variability between events.
- AngleX. This field contained the horizontal access in degrees in a scale from 0-360 degrees. Zero degrees equated to a shot to the front of the vehicle; 90 degrees equated to a shot on the vehicle right side. This was also modeled using a triangular distribution to allow for variability between events.
- AngleY. This field contained the vertical access in degrees in a scale from 0-180 degrees. Zero degrees equated to a shot to the top of the vehicle; 180 degrees equated to a shot to the bottom of the vehicle. As with AngleX, this was modeled using a triangular distribution to allow for variability between events.
- PointOfAimX, PointOfAimY, PointOfAimZ. These field contained the coordinates on each platform of the intended impact of the ballistic round. Each modeled vehicle had a coordinate system with the origin set at its left rear corner. All distances were measured in millimeters. We used a triangular distribution to specify these coordinates to vary the impact of each round.

B.2.2.2 Capability_Assessment_Init [5].

This table was used for development and validation purposes only, not for normal simulation runs. It contained the field Entity_Name, child of the same field in the List_Entity_Name [3] table, and 19 additional fields for the assessment of the capability state at the beginning of the simulation: Mobility 1 through 10; Lethality 1 and 2; Communication 1 and 2; and Protection 1 through 5. The table was connected to the ExtendSim DataInit block to establish starting capability conditions for platforms when the simulation was not integrated with MUVES.

B.2.2.3 Entity_Data [17].

This table was the single most referenced table in the database by the blocks in the simulation to govern the behavior of the entities. In addition to the Entity_Name field, it contained 22 additional fields, 11 of which were children of fields in one of the list tables.

- Entity_Type. A child of the field of the same name in the List_Entity_Type [1] table, it was used to identify the vehicle (platform) types and unit types for non-vehicle entities.
- Init_Op_KM. This field identified the number of operational kilometers that each entity had driven prior to the beginning of the simulation. It was modeled as a triangular distribution with a maximum value of the mean number of kilometers before system aborts for an AMPV, 117 kilometers for this study.
- SP_Time. This field contained the start time in simulation minutes for each entity according to the TOEL.
- Route_TRM. A child of the field of the same name in the List_Routes [18] table, it was used to identify which route each entity was to travel during the Extended Tactical Road March phase.
- BallisticEventProb. This field established the probability, at each event point, of the entities being involved in a ballistic interaction. For this study the value was set at 0.25 for all entities.

- TRM_Loiter. A child of the field of the same name in the List_LoiterLoc [26] table, it identified which entities would temporarily stop near the city of Dalhart during the Extended Tactical Road March phase.
- Axis_DA_Exp. A child of the field of the same name in the List_Routes [18] table, it identified which route each entity would travel during the Deliberate Attack Exploitation phase.
- DAEScreenMO1. This field identified the order of march the entities would follow screening route during the Deliberate Attack Exploitation phase. An entry of 9999 was used for any entity not using this route.
- DAERedMO1. This field identified the order of march the entities would follow on Axis Red during the Deliberate Attack Exploitation phase. An entry of 9999 was used for any entity not using this route.
- DAEWhiteMO1. This field identified the order of march the entities would follow on Axis White during the initial portion of Deliberate Attack Exploitation phase. An entry of 9999 was used for any entity not using this route.
- CavDAEMsn2. A child of the field of the same name in the List_Cav_Mission [35] table, this field identified the route and cavalry mission for each cavalry entity during the Deliberate Attack Exploitation phase. An entry of NA indicated that this field was not applicable for a given entity.
- DAEWhiteMO2. This field identified the order of march the entities would follow on Axis White during the later portion of Deliberate Attack Exploitation phase. An entry of 9999 was used for any entity not using this route.
- DAERedLDDel. This field established a delay time for entities on Axis Red prior to them proceeding out of Assembly Area Logan. The amount of delay enabled the alignment of the entity movement to the LD times prescribed by the TOEL.
- DAERedCP3TaskOrg. A child of the field of the same name in the List_TaskOrg1-7 table, this field identified which team each entity of 1-7 CAB would be task organized to during the Deliberate Attack Exploitation phase.
- DAURedDaytonRel. A child of the field of the same name in the List_Routes [18] table, this field identified which units would remain the vicinity of Objective Dayton or would move with TF2-7 on Axis Red during the Deliberate Attack Urban Environment phase.
- DAURedMO1. This field identified the order of march the entities would follow on Axis Red during the early portion of Deliberate Attack Urban Environment phase. An entry of 9999 was used for any entity not using this route.
- DAURedMO2. This field identified the order of march the entities would follow on Axis Red during the middle portion of Deliberate Attack Urban Environment phase. An entry of 9999 was used for any entity not using this route.
- AxisDAU_Red_PA1. A child of the field of the same name in the List_Routes [18] table, this table identified which entities on Axis Red were to stop at Position Area 1 or to continue without stopping during the Deliberate Attack Urban Environment phase.
- DAURedMO3. This field identified the order of march the entities would follow on Axis Red during the final portion of Deliberate Attack Urban Environment phase. An entry of 9999 was used for any entity not using this route.
- AxisDUA_Red_End. A child of the field of the same name in the List_Routes [18] table, this table identified which of two possible branches that entities travelling on Axis Red were to take during the final portion of the Deliberate Attack Urban Environment phase.

- Entity_Category. A child of the field of the same name in the List_Entity_Category [2] table, this table identified each entity as either a vehicle or unit. This field was referenced by the Repair Event block to direct non-vehicle entities to exit the block without undergoing repairs.
- Platform_Type. A child of the field of the same name in the List_Platform_Type [45] table, this table directed the path of the entities through the appropriate paths of the platform assessment constructs.

B.2.4 Entity_Type_Data [16].

This table contained five fields that were referenced by blocks in the model to access data about entities that was specific to its platform type.

- Entity_Type. A child of the field of the same name in the List_Entity_Type [24] listed the possible vehicle (platform) types and unit types for non-vehicle entities and served as the record index for the table.
- Entity_Category. A child of the field of the same name in the List_Entity_Category [2] table, this table matched the Entity_Category, vehicle or unit, with the Entity_Type.
- MKMBSA. This field established the Mean Kilometers Before System Aborts for each entity type and was modelled as an exponential distribution with a mean of 117 KM. This field was used in conjunction with the Init_Op_KM of the Entity_Data [17] table for reliability modeling of the platforms.
- Recovery_Time. This field established the time required for recovery of a vehicle that lacked Mobility. A triangular distribution with a mean of 30 minutes was used for this study.
- MTTR. This field established the Mean Time To Repair for an entity requiring it due to a capability shortfall. Modeled as a lognormal distribution with a mean of 54, this field was referenced by the Repair Event constructs of the simulation.

B.2.3. Output Tables

The output tables were the tables that contained the assessments of platforms and units collected during the simulation.

B.2.3.1 Capability_Assessment [29].

This table contained the running capability assessment of all platforms, entities 1-59, throughout the simulation. The table index field was Entity_Name, a child of the field of the same name in the List_Entity_Name [3] table. The remaining fields contained the assessment of the entity based on the 19 capability states: Mobility 1 through 5; Lethality 1 and 2; Communication 1 and 2; and Protection 1 through 5.

B.2.3.2 TaskCapCol_#### [14, 15, 41, 42, 44, 47]

Where the #### was replaced by the designation of the appropriate studied events from the Deliberate Attack Exploitation: P104, P107, P108, P110, P114, or P114. These tables contained the collective task assessment for each of the 12 aggregated units in the simulation. The table index field was Unit_Name, a child of the field of the same name in the List_Unit_Name [6] table. The tables contained the assessment for tasks that were relevant for the interactions at the studied event. The collective tasks were designated as CTask1 through CTask12; each field's properties contain the complete task name which could be viewed by opening the field properties or hovering the mouse over the field and pop-up would appear containing the full task name; for example, CTask1 was ART 6.6.3.1 Provide a Screen. After a replication, the simulation exported the data in the table to an MS Excel spreadsheet.

B.2.3.3 TaskCapPlat_#### [7, 9, 10, 11, 13, 46]

Where the #### was replaced by the designation of the appropriate studied events from the Deliberate Attack Exploitation: P104, P107, P108, P110, P114, or P114. These tables contained the platform task assessment for each of the 59 platform entities in the simulation. The table index for this field was Entity_Name, a child of the field of the same name in the List_Entity_Name [3] table. The tables contained assessment for task that were relevant for the interactions at the studied event. The platform tasks were designated as PTask1 through PTask12; each field's properties contain the complete task name which could be viewed by opening up the field properties or hovering the mouse over the field and pop-up would appear containing the full task name; for example, PTask 1 was ART 6.4.3.4 Establish Observation Posts. After a replication, the simulation exported the data in the table to an MS Excel spreadsheet.

B.2.4 Status

The Status tab contained one table named Entity_Status [19] used in reliability event calculations.

B.2.4.1 Entity_Status [19].

In addition to the table index, this table contains two additional fields Op_KM and KM_Next_Failure. This table was populated by blocks in the simulation that calculated the number of kilometers the platform had travelled since its last system failure and the number of kilometers until the next system failure would occur. The table index for this field was Entity_Name, a child of the field of the same name in the List_Entity_Name [3] table.

B.2.5 Location_Data

This tab contained one table used by the simulation to determine the behavior of entities moving at different time and locations during the simulation.

B.2.5.1 Route_Segment_Data [22].

This table contained three fields that the Segment_Transport constructs called upon to determine entity travel time and speed. The RouteSegment field, a child of the field with the same name in the List_Route_Segment [23] table, identified individual route segments throughout the simulation designated by the beginning and end EventPoint_Name separated by a dash; for example, Route Segment H-M spanned the distance between event points H and K. R_Distance identified the total length of the route segment in kilometers. R_Speed identified the speed that each entity travelled on that route segment.

B.2.6 Log

The simulation contained three primary logs each used to capture data relevant to the studied interaction events.

B.2.6.1 Log_MaintRecovery [25]

This log was used to capture relevant data about reliability failure events. The field RecoveryStart_Time identified when each reliability event occurred. EventPoint_Name, a child of the field with the same name in the List_EventPoint_Name [20] identified the event point for the interaction. Entity_Name and Entity_Type, children of fields of the same name in the List_Entity_Name [3] and ListEntity_Type [1] tables, identified which entity was involved in the interaction and its type. The Recovery_Time field identified how it took to recover the entity if recovery was required; only platforms that lost Mobility required recovery.

B.2.6.2 Log BallisticEvent [27]

This log was used to capture relevant about events that included ballistic interactions. This table was setup almost identically to the Log_MaintRecovery [25] table having the fields EventPoint_Name, Entity_Name, and Entity_Type. The BallisticEventStart_Time field identified when the ballistic interaction occurred and the Recovery_Time field identified how it took to recovery the entity if recovery was required.

B.2.6.3 Log_RepairEvent [32]

This log was used to capture relevant about events where platforms were repaired. As with the other two logs, it contained the fields EventPoint_Name, Entity_Name, and Entity_Type. The RepairEventStart_Time identified when repairs to the platform commenced and the Repair_Time identified the overall repair time taken during the event.

B.2.7 DLL

This tab contained tables used by ExtendSim containing data related to ExtendSim's communication with MUVES via the API. The first three tables listed below contained log data of the interaction events while the other two tables contained data detailing the communication messages between ExtendSim and the API.

B.2.7.1 BallisticEventDLLData [4].

This table contained the ballistic event parameter data for each of the ballistic interactions as well as the capability assessment of the platform following the interaction. The table index was the field Entity_Name, a child of the field of the same name in the List_Entity_Name [3] table. The next several fields were structured identically to the BallisticEventParanterData [31] table but containing the specific ballistic event parameters for each interaction: threat munition type involved, the weapons system, the angle the munition impacts the platform, and the impact point (point of aim) on the vehicle. ExtendSim passed this data through the API to MUVES. MUVES returned through the API the capability assessment of the platform based on the 19 capability states: Mobility 1 through 5; Lethality 1 and 2; Communication 1 and 2; and Protection 1 through 5, which were written by ExtendSim into the database in the appropriate entry for each interaction.

B.2.7.2 aintFailEventDLLData [28].

Like the BallisticEventDLLData [4], the table index was the field Entity_Name, a child of the field of the same name in the List_Entity_Name [3] table. The field Fail_Types, a child of the field of the same name in the List_Fail_Types [34] table, the type of system failure the platform suffered because of the reliability failure in the interaction. The remainder of the fields contained the capability assessment of the platform, returned by MUVES through the API, following the interaction against the 19 different capability levels.

B.2.7.3 RepairEventDLLData [30].

This table's index was also the field Entity_Name. The remainder of the fields contained the capability assessment of the platform following the repair event, returned by MUVES through the API, against the 19 different capability levels. Since repair events were designed to "completely repair" all shortcomings, the resulting table displayed "1" in every capability category.

B.2.7.4 DLLConnectionData [38].

This table was created by the MUVES block in ExtendSim during the individual replications and verified that all the procedures within the MUVES block to API connection were valid.

B.2.7.5 DLLConnectionLog [39].

This table was created by the MUVES block in ExtendSim during the individual replications and enumerated all of the message traffic that passed through the API.

B.3 Application Program Interface (API)

The ExtendSim to MUVES API used a Data-link library (DLL) - ExtendSimMUVES.dll – installed in the Extensions\DLLs sub-folder of the ExtendSim folder. The DLL was written in C++. To connect ExtendSim to the API there were four custom blocks written to facilitate the transfer of data from the ExtendSim constructs to the DLL. Each type of interaction event had a custom ExtendSim custom block (named Ballistic Event, Maintenance Event, and Repair Event respectively) that sent a message initiating the API sequence. Each custom block sent the required data about the interaction to another custom ExtendSim block named MUVESManager. The MUVESManager block passed the data through the DLL to MUVES. Data returning from MUVES passed into the MUVESManager which wrote the data to the Capability_Assessment [29] table and the appropriate log and DLL tables for the event in the ExtendSim database.

There were four times when exchanges between ExtendSim and MUVES using the API occurred: Initialization, a Ballistic event, a Reliability event (Maintenance Failure Event), and Repair event. During Initialization, ExtendSim called MUVES through the API to initialize all capability states for each entity. MUVES then passed the data back through the API, setting all capability state values (Mobility 1 through 5; Lethality 1 and 2; Communication 1 and 2; and Protection 1 through 5) for the 59 platform entities to a value of 1, fully capable. ExtendSim then wrote these values to the Capability_Assessment [29] table for all entities.

The data message traffic through the API was similar for each of the messages. In all three messages, ExtendSim passed the EntityID of the platform involved in the event through the API. For a Ballistic event, ExtendSim also passed the shot parameters of the ballistic engagement which included the munition type, range, angle of the trajectory between the point of launch and the target, and point of aim on the targeted platform. MUVES would then pass back through the API to ExtendSim the revised capability state values for the platform involved in the interaction. For a reliability event (Maintenance Failure Event), MUVES additionally returned the specific sub-system that failed leading to the degraded capability assessment for the impacted platform (Figure 64).

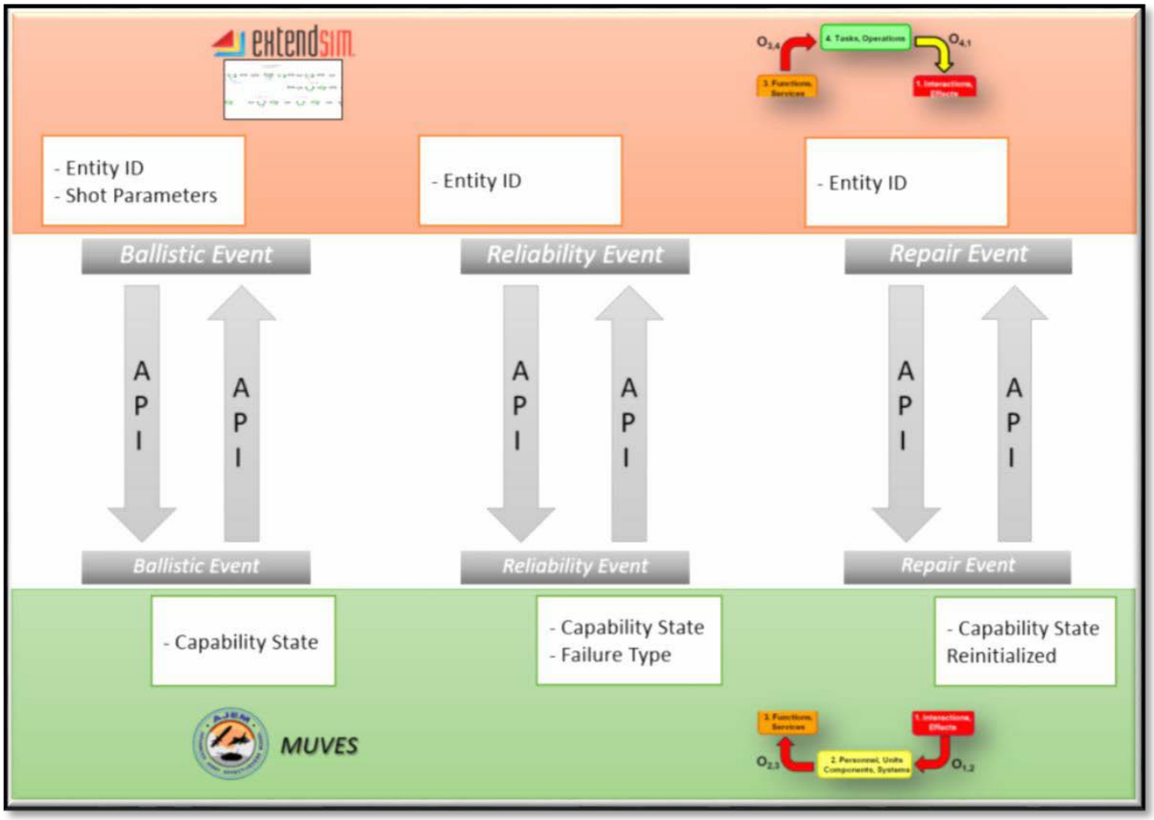


Figure 64 MMF Dynamic Assessment Suite API

DISTRIBUTION LIST

Organization

U.S. Army Materiel Systems Analysis Activity
AMXAA-GL/Reports Processing
392 Hopkins Rd.
Aberdeen Proving Ground, MD 21005-5071

Defense Technical Information Center ATTN: DTIC-O
8725 John J. Kingman Rd Fort
Belvoir, VA 22060-6218